An Empirical Modelling Approach To Software System Development in Finance: Applications and Prospects

By

Soha Maad

Thesis

Submitted to The University of Warwick

In partial fulfillment of the requirements

For admission to the degree of Doctor of Philosophy

Department of Computer Science

University of Warwick

March 2002

Contents

List of Tables

List of Figures

In Memory Of My Father

Acknowledgements

I would like to deeply thank my supervisor, Meurig Beynon, for supporting my research work and encouraging me to pursue my research objective. I would like also to thank Steve Russ for his comments on my research work and focus. Many thanks also to Graham Nudd for advising me throughout my research.

Further thanks goes to Mike Holcombe for his constructive comments during the viva and to Steve Russ for his useful help on the final version of the thesis.

The financial concepts and their technical implementation presented in this report has benefited from the discussions with, and attendance of MSc and doctoral workshops of Abhay Abhiankar, Peter Corvey , Marcus Miller, Nick Webber, J. Harrison, Nick Taylor, Mark Taylor, Van Sebastian Strien, and Mark Stewart at Warwick Business School and the Department of Mathematics and Statistics at the University of Warwick. Many thanks also to my colleague, Keng-Yu Ho, at Warwick Business school, for sharing with me the technical part of his doctoral research study. Thanks also to Samir Garbaya, at the Robotics Laboratory in Paris “Laboratoire de Robotique de Paris”, for the joint publication work on virtual reality.

A great thanks is directed to the previous Secretary General of the Council of Ministers in the Lebanese Republic, Honorable Hisham Chaar, for supporting my scholarship from the Islamic Development Bank (IDB) in Saudi Arabia. Many thanks as well to all the staff of the IDB Merit Scholarship office for their careful follow up of my scholarship.

A warm thanks is directed to my mother, and sisters Maria, Nihal, Eva, and my little dear sister Hoda, and my Aunts Racha Nahas and Nabil Nahas for their moral support to me during this challenging period.

Finally I thank God for helping me to do this study and I highly confess that no matter how human knowledge grows it stays limited.

Declarations

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

The perspective on the integration of financial systems in this thesis has been published in [BM99]. The various aspects of building a web-based environment for virtual collaboration appeared in [BM00]. The timetabling model was considered in [BWMRR00]. The integration of e-commerce and ERP applications was researched in [Maa99]. The EM and VR models of the monopoly dealer simulation were introduced in [MBG01]. The financial analysis in the Ho case study can be found in [AH01].

Abstract

The financial industry is witnessing major changes. The financial enterprise is undergoing major business process renewal accompanied with the introduction of new technologies including electronic commerce; the financial market is shifting from an old to a new trading model that introduces major structural changes to the market and new roles for market participants; investment offers access to ever larger repositories of financial information and a wider choice of financial instruments to fulfill rising needs and expectations. In all these developments, there is a central role for human intelligence that can potentially influence the pattern of change and direct appropriate decisions in adapting to change. There is also a vital need for computer-based technology to support this human activity.

The relation between human and computer activities in classical models for computer-based support is characterised by rigidity and framed patterns of interaction. The emphasis in such models is on automation, not only in respect of routine trading operations, but even of the role of market participants. An alternative culture is emerging through the use of advanced technologies incorporating databases, spreadsheets, virtual reality, multi-media and AI. There is an urgent need for a framework in which to unify the classical culture, in which mathematical financial modelling has a central place, with the emerging culture, where there is greater emphasis upon human interaction and experiential aspects of computer use. This thesis addresses the problem of developing software that takes into account the human factor, the integration of the social and technical aspects, human insight, the experiential and situated aspects, different viewpoints of analysis, a holistic rather than an abstract view of the domain of study, cognitive rather than operational activities, and group social interaction. The ultimate aspiration for this work is to transform the computer as it is used in finance from an advanced calculator to an 'instrument of mind'.

Meeting the challenges of software support for finance is not only a matter of deployment, but also of software system development (SSD): this motivates our focus on the potential applications and prospects for an Empirical Modelling (EM) approach to SSD in finance. EM technology is a suite of principles, techniques, notations, and tools. EM is a form of situated modelling that involves the construction of artefacts that stand in a special relationship to the modeller's understanding and the situation. The modelling activity is rooted in observation and experiment, and exploits the key concepts of observables, dependencies and agency. The thesis extends the major findings of Sun (1999), in respect of the essential character of SSD, and its contextual and social aspects, by considering its particular application to the finance domain.

The principles and qualities of EM as an approach to SSD are first introduced and illustrated with reference to a review of relevant existing models. The suitability of EM as a framework for SSD in finance is then discussed with reference to case studies drawn from the finance domain (the financial enterprise, the financial market, and investment). In particular, EM contributes: principles for software integration and virtual collaboration in the financial enterprise; a novel modelling approach adapting to the new trading model in the financial market; computer-based support for distributed financial engineering; and principles for a closer integration of the software system development and financial research development activities. This contribution is framed in a Situated Integration Model, a Human Information Behaviour Model, an Open Financial Market Model, a framework for distributed financial engineering, and a situated account of the financial research development cycle.

Abbreviations

AI – Artificial Intelligence

EM – Empirical Modelling

CRM – Customer Relationship Management

CAPM – Capital Asset Pricing Model

APT – Arbitrage Pricing Theory

OOP – Object Oriented Programming

XML – Extensible Markup Language

ERP – Enterprise Resource Planning

P/E – Price to Earnings ratio

ISM – Interactive Situation Model

EMH – Efficient Market Hypothesis

OLS – Ordinary Least Square Regression

EMF – Empirical Modelling Framework

VR – Virtual Reality

DEM – Distributed Empirical Modelling

ISM – Interactive Situation Model

CD – Cognitive Dimensions

IA – Information Artefact

SOM – Self Organized Maps

SSD – Software System Development

RE – Requirement Engineering

SPORE – Situated Process of Requirement Engineering

DSS – Decision Support System

GDSS – Group Decision Support System

HCI – Human Computer Interaction

VR – Virtual Reality

ODM –Open Development Model

BPM – Business Process Modelling

BPR - Business Process Re-engineering

SE – Software Engineering

OOSE – Object Oriented Software Engineering

NPV – Net Present Value

IRR – Internal Rate of Returns

DOT – Distributed Object Technology

GO – Generic Observable

Jam – Java Maintainer Machine

API – Application Interface

Chapter 1

Introduction

1.0 Overview

This chapter is an introductory one, expressing my research motivation and aim. It outlines the content of subsequent chapters, and reveals my major research contributions and achievements

1.1 Research Motivation and Aim

This research aims to explore computer-based technology support for the financial industry. My research interest stems from my awareness of the importance of the role played by finance in the prospect of humans and nations and the role of computer-based technology as a driving force in all business sectors including the financial one. Computer-based technology is becoming an integral part of our life, and its potential use in supporting our cognitive and routine activities is of critical importance. It is impossible to ignore the role for computer-based technology in our daily activity, but choosing an appropriate technology solution is a challenging task.

The financial industry is witnessing major changes manifested in the introduction of new financial instruments, major structural changes in the financial markets, increasing financial markets integration, and new roles for market participants [Lan99, Ode00, RB00]. Computer-based technology has a central role in motivating interest in change and suggesting paradigms for its application [FRKBCJ00, Dav93, Dav96]. This thesis identifies some key issues for the use and application of computer-based technology in the financial industry at the institutional, market, and investment levels. The following paragraphs elaborate on these issues at these three levels.

Key issues of the application of computer-based technology in finance

At the ***institutional level***, computer-based technology plays an important role in leading the structural institutional change to gain competitive advantages in a global market [Dav93]. *Software integration* [BM99], and *virtual collaboration* emerge as key concerns in the financial enterprise.

At the ***market level***, computer-based technology plays an important role in *supporting the shift from an old to a new trading model* [Lan99]. The old trading model depends on a physical place, a central role for financial inter-mediation, and limited geographical spread of trading activity. In the new trading model, an appropriate use of computer-based technology aims at delivering straight through processing (STP), facilitating cross border trading, supporting market integration, reshaping the role of different market participants, re-engineering financial trading systems, and providing intelligent management and delivery of trade information, reporting, and investment services.

At the ***investment level***, *financial engineering*, and *the finance research development cycle* are increasingly relying on computer-based technology and the power of the computer to support intelligent human activity.

Challenges facing the application of computer-based technology in finance

Acknowledging the significant role of computer-based technology in the advance of the financial industry is important, but recognizing its limitations in delivering the intended support is essential in order to overcome these limitations and achieve the targeted goals.

Major challenges are faced in the application of computer-based technology in the financial industry. This thesis identifies the key problem in meeting these challenges as finding a suitable framework for software system development (SSD) for modern finance.

It is convenient to view software system development in finance as represented by two cultures:

(1) a traditional culture[1] that relies on orthodox techniques that frame the software system development into rigid stages. These approaches to SSD typically rely on predefining and fixing a boundary for the software system to be developed, there is an increasing awareness of the need for a more flexible boundary that can grow as the development proceeds[2];

(2) an emerging culture that involves "ad hoc" uses of spreadsheet, databases, virtual reality VR, artificial intelligence, multi-media, etc.

The emerging culture in which advanced databases, spreadsheet, virtual reality, multi-media, and artificial intelligence are getting applied is quite different from the classical computing culture with its strong mathematical and logical foundations. There are potential benefits in finding a suitable framework for studying both the traditional and the emerging computing culture to SSD in diverse domains, including finance. This is particularly relevant because of the prominence of mathematical and statistical models in the finance domain and the importance of unifying the logical / mathematical with the informal experiential aspects. More generally, there is a related need for SSD principles that help to integrate automated processes with activities directed by human judgement, discretion, and observation of the world. In this respect, traditional SSD is not well suited to the demands of modern finance.

[1] This refers to method-based approaches to software system development including traditional waterfall methods [SS95, STM95], object-oriented methods [Boo94, CY90, JCJO92]. Method based approaches to software system development provides formally-defined methods, techniques and tools that can in principle guide the development of software systems in a systematic and cost-effective fashion.

Moreover, whilst the concepts and techniques of the emerging culture helps to address various aspects of the new SSD agenda, there is arguably no satisfactory foundational framework[3] within which to gain a holistic view. Indeed, the impact of new technologies and techniques has to some extent been a negative one, compounding the problem of integration by extending the catalogue of disparate technologies. This motivates our interest in a paradigm shift in SSD practices for the finance domain.

Motivation for EM

Empirical Modelling (EM) technology (a suite of principles, tools, techniques, and notations) aims to achieve greater integration of the human and computational activity by adopting novel software system development practices[4]. Its key concepts are agency, dependency, and observation. To some degree, these concepts are represented in current SSD practices – for instance in agent-based systems, observer-like objects in object oriented development and spreadsheets, but the way in which they are deployed in EM is distinctive, and invests them with an ontological status that is quite unlike that associated with classical models of computation. EM involves situated activity in which the roles of the modeller, the artefact, and the situation are inseparably linked, and leads to the construction of computer-based artefacts that stand in a special relation to the modeller's understanding and to the situation to which they refer.

EM technology emphasizes the importance of a broad foundation for computing capable of embracing, when possible, disparate technologies in an open development environment. Throughout its evolution Empirical Modelling technology has shifted focus from programming an application to modelling a domain. EM technology provides a modelling framework that precedes Object Oriented Modelling in identifying entities and their reliable patterns of behaviour suitable for object and corresponding methods abstraction [BJ94]. It proposes new foundations for artificial intelligence that take into account experiential knowledge in developing computer-based artefacts [Bey99]. It promises support to VR technology at the analysis and design level to take into account the social aspect of a virtually mediated environment [MBG01]. It provides a framework that complements formal representation and rigid symbol based modelling with experience based modelling favouring semi automated activities [BRR00-1, BRR00-2, CRB00].

---

[2] For instance, there are several approaches that regard software system development as a 'situated' activity in the sense used by [Suc87] and that aim at reconciling the social and technical aspect in software system development [Fit96, Gog94, HKL95, HKN91, Flo95, CS90, Mum95, AF95, Sun99].

[3] Though proposals such as Checkland's Soft System Methodology (SSM) [CS90] and other systems theory approaches are in this spirit.

The thinking behind Empirical Modelling Technology stands in a particularly interesting relation to critiques of traditional empirical approaches[5], both from a philosophical perspective, and as they apply to computer science topics such as software engineering and AI. These include: the philosophical stance of Radical Empiricism[6] of William James, the views of Fred Brooks on Software engineering theory and practice[7], Kirsh's critique of the logicist[8] approach in Artificial Intelligence, and the founder[9]'s view on establishing a new computational framework capable, when possible[10], of embracing disparate principles, theories, and foundations.

Empirical Modelling technology motivates a new software culture [Sun99, Bey01] that potentially supports a stronger relationship between computer-based technology and various activities in social application domains. In particular, it addresses the following fundamental issues of SSD identified in [Sun99]:

(1) SSD is a human activity that needs technical support in order to enhance its application to social domains;

(2) SSD is highly associated with its context, which must be considered in a situated manner;

(3) human agents are the most important dimension in SSD;

(4) interpersonal interaction is the most crucial activity in SSD; computer-based support plays an enabling role in facilitating human agency and promoting interpersonal interaction.

The essential character of SSD in EM suggests a clear distinction between SSD in EM and traditional SSD. The latter is a process framed into rigid stages, a closed boundary system, and a preconceived functionality for an end product to be delivered at the end of the SSD process. On the contrary SSD in EM consists of a collection of situated activities that arise in the construction and use of the required system in the real world. This situated activity takes into account both technical and social aspects and directs a central focus on the interaction

[4] Refer to amethodological SSD proposed in [Sun99].

[5] As characterized by the relationship between what is interpreted as "given by theory" and "given by observation" in an experimental context.

[6] It is characteristic of Radical Empiricism that relationships in the world, that are traditionally attributed to theory, are interpreted as empirically given.

[7] There is no "silver bullet": no single software engineering methodology that can deliver a complete solution to complex problems in the real world domain.

[8] This is as characterized by the physical symbol system hypothesis introduced by Newell and Simon (1976).

[9] The founder of the Empirical Modelling project is Meurig Beynon.

[10] In discussing Empirical Modelling as a new foundation of artificial intelligence, Beynon (1999-2), argues that there is no possibility of merging the logicist and non-logicist approaches in artificial intelligence.

between human agents involved in the SSD activity and the product-under-development. This product under development is represented by a computer-based model that reflects an evolving software system with dynamic rather than rigid boundaries. Deriving useful systems from an EM approach to SSD necessitates an appropriate circumscription of the model once a desired functionality can be relied upon.

The principal claim of the thesis

The principal claim of the thesis is that EM technology promises to offer computer-based support to software system development (SSD) that is better adapted for the financial industry at the institutional, market, and investment level. EM can potentially help to address the need of finance in the following ways:

At the institutional level, by establishing principles and foundations for software system development meeting integration and virtual collaboration needs.

At the financial market level,

(1) by providing computer-based support to software system development adapted to the needs of the new trading model;

(2) by establishing principles that support qualitatively different trading patterns by accounting for different factors including: a richer set of observables shaping and being shaped by the trading environment, different modes of interaction and behaviour of market participants, and different insights for strategic decision making;

(3) by observing the virtual trading activity from different perspectives (technical, social, and human).

In investment research and analysis,

(1) by enabling a more intimate relationship between the software system development activity and the finance research development activity;

(2) by supporting distributed modelling in software system development for financial engineering.

Strategy for supporting the claims

Empirical Modelling technology draws on well established principles, techniques, and tools that introduce a radical shift in perspective in software system development practices. We will argue that EM has strategic implications of relevance to finance because (in principle) it:

a) supports an open, human centred, amethodological, context-dependent, situated, and user-centred approach to software system development

With the complexity and continuous change in the financial industry, adopting rigid approaches to software system developments might fail to cope with the change and incur the

financial industry enormous re-engineering costs. An EM approach to software system development provides greater flexibility and adaptability to change.

b) *maintains a semantic relationship between the computer-based artefact and its real world referent*

In modelling an application domain, EM technology establishes a semantic relationship between the computer-based aretefact and its real world referent. This motivates a re-thinking of the software system development cycle, the requirement engineering phase in software system development, and the system's boundary.

c) supports experiential knowledge construction in software system development

The finance literature reveals conflicting approaches and theories (e.g. Arbitrage Pricing Theory (APT) vs the Capital Asset Pricing Model; Fundamental vs Chartist financial analysis, the random walk and Efficient Market Hypothesis vs forecasting future price indicators based on past indicators; etc). This motivates experiential knowledge construction in software system development for the finance domain, and is supported by Bryan's view on mathematical financial modelling:

" *Financial modelling*[11] *is emphatically not an academic process, and past experience has shown elegant mathematics and sophisticated programming as more likely to lead to failure rather than forming a pathway to success. The importance of models being developed in close conjunction with (if not by) the user cannot be over-emphasised*" [Bry82].

d) supports interpersonal communication in software system development

Group social activity is central at all finance levels (financial enterprises, markets, and investment). Financial markets can be viewed as a social network grouping different market participants (brokers, dealers, investors, specialists, etc..). The financial enterprise is a social network comprising top management, regulatory and supervisory authorities, and operational entities. The financial analysis and investment activity is a group activity involving a network of academic researchers and practitioners. This motivates greater account for the group social activity in software system development for the finance context. Interpersonal interaction in

[11] Financial modelling is the process of creating a *picture* or *financial model* of a financial scenario, where the model represents a simplification of the real situation [Bry82]. Financial models can be of two types: analogue models (physical representation of objects or situation that does not actually look like the real thing, examples include diagrams and charts), or mathematical models (models composed

the software system development activity for finance and in collaborative group work activity related to finance poses great challenges to emerging communication-based technologies such as the internet. This is mainly due to the complexity of interpersonal interaction and the difficulty of providing it with the appropriate computer-based support [Son93]. EM proposes a computer-based support for interpersonal communication that takes into account various modes of interaction and the role of agency in communication, and relies on the communication of definitive scripts as a medium for sharing artefacts and insight.

e) supports software system development with explanatory modelling of application domains
Traditionally, mathematical models and natural language have been used to represent structured knowledge of the finance domain. Financial trading is an example of a context where mathematical abstraction may fail to convey the behaviour of different trading parties, trading signals, and trading systems. There is a rising need for the investigation of new principles to supplement traditional mathematical modelling in modelling complex trading environments. The ability to forecast the impact of a new technology or structural change prior to its introduction, and to investigate an optimal financial market structure and organization, need more than traditional financial modelling techniques and call for cognitive and experience based modelling. In this connection, business process modelling, business process re-engineering, and analysis of virtual social trading networks are important considerations.

f) motivates new relationships between the software system development activity and diverse real world activities that are not inherently computer-based
By considering the situated nature of the software system development activity new relationships between SSD and various activities in social application domains emerge. Examples are relationships between the finance research development activity and the SSD, and business process modelling and software system development.

Implications a) through f) are wide-ranging and are not specific to financial applications. The EM tools are not sufficiently mature to allow us to demonstrate these implications in practical applications in finance at present, especially on the scale that is appropriate in industrial case studies. Previous case studies using EM tools do, however, supply proof-of-concept evidence

---

of symbols and mathematical relationships that represent a real situation). Financial models can be also classified by the way they are used, such as for simulation, optimisation, or forecasting [Hon99].

in relation to a) through f). On this basis, the thesis makes it plausible that EM can deliver a) through f) by:

1) discussing the wider agenda of computational activity in addressing the technical and strategic demands of the applications of computer-based technology in finance: this involves identifying these demands and exposing the challenges current technologies face in meeting them. It also motivates a paradigm shift at the computational level (cf. chapter 2).
2) introducing EM technology as a suite of principles, techniques, notations and tools with particular reference to simple case studies drawn from the finance domain. The chosen models enable us to discuss the significance of agency, dependency and observables in relation to particular financial applications (cf. chapter 3).

3) highlighting the distinctive qualities of model building in EM that motivate a paradigm shift at the computational level (cf. chapter 3). These qualities include:
   - the focus on state as experienced
   - knowledge construction using an artefact
   - the support of collaborative relationships in distributed modelling
4) reviewing the broad implications of EM for software system development meeting the wider agenda for computational activity: This involves the analysis of relevant history of EM, in particular with reference to previously developed models that demonstrate the potential broader implications of adopting an EM paradigm (cf. chapter 4). The case studies considered are:
   - A *railway accident model* that demonstrates knowledge re-construction of a historical event in a distributed environment that reveals the complexity of social communication and the different possible construals of a historical event.
   - A *virtual electronic laboratory model* that demonstrates the complexity of the technical support needed for group social interaction and highlights the situated nature of this interaction.
   - An *attribute explorer* that supports the visual amethodological exploration of data sets. This exploration helps in identifying complex patterns of relationships between data sets.
   - A *timetabling model* that illustrates a novel mode of decision support that allows close integration between manual and automated activity.

5) by developing proof-of-concept models that illustrate how EM might eventually deliver new solutions to problems in the finance domain different in character from solutions delivered by existing computer-based support. Relevant issues to be examined include:
   - *Software system development* in the financial enterprise with particular emphasis on integration and virtual collaboration (cf. chapter 5).
   - *Software system development* for the financial market with particular reference to open and explanatory modelling activity and the use of artefacts to construe complex application domains (cf. chapter 6).
   - Distributed modelling in *software system development* for financial engineering (cf. chapter 7).
   - The relationships between the *software system development* activity and the finance research development activity (cf. chapter 8).

The above issues 1) .. 5) are considered in the referenced chapters. The outline of each of these chapters follows.

1.2 Thesis Outline

The principal aim of this thesis is to investigate the application of an experience based technology, "Empirical Modelling (EM)" technology, to the finance domain. The research is framed within a web based environment[12]. Several case studies from finance are considered, these include: software integration and virtual collaboration in the financial enterprise, building virtual environments for financial trading, computer-based support for financial engineering, and computer-based support for a typical finance research development cycle.
The rest of the thesis consists of eight chapters that are organized as follows:

Chapter 2 frames key issues for the application of computer-based technology in finance. It argues that finance makes technical and strategic demands on computing that pose great challenges to current tools and technologies. This leads us to the identification of a "wider agenda for computing" that motivates a paradigm shift at the computational level.

Chapter 3 introduces EM technology as a suite of key concepts, techniques, notations, and tools. It highlights the distinctive qualities of model building in EM that can potentially meet the technical demands of finance identified in chapter 2.

Chapter 4 discusses the broad implications of EM on the wider agenda of computing. It examines the nature of the paradigm shift in SSD associated with EM, and explores the implications for the strategic demands of finance identified in chapter 2. The latter are illustrated with reference to various previous case studies in EM. These case studies show in particular that EM can potentially lead to a closer integration between the software system development activity and activities in real world domains, a theme that is developed in the remainder of the thesis.

Chapter 5 considers the prospects for EM technology in the financial enterprise, and in particular proposes a new framework for software system development for integration and for virtual collaboration. Software integration and virtual collaboration are examples of two software development activities where people, devices, and programmable components should integrate more coherently to attain their objective. This entails viewing a system from many different perspectives, taking into account the social and the technical aspects as well as their coherent integration. Devising a solution meeting all requirements is hardly possible. The solution is situated in nature and grows with growing needs. The technical and social aspects of software integration are discussed with reference to a Situated Integration Model (SIM). The needs and requirements for integration are motivated with the example of the integration of ERP and e-commerce applications. In software system development for virtual collaboration, EM emphasizes the importance of considering human information behaviour (HIB) with reference to its associated context, situation and social network, as characterized by Sonnenwald in [Son99].

Chapter 6 proposes a new modelling approach for software system development for the financial market adapted to the new trading model. The proposed Open Financial Market Model (OFMM) is characterised by the openness of the modelling activity and its use for requirement engineering throughout the entire software system development. The technical implementation and practical application of the OFMM is considered with reference to a case

[12] This web-based environment includes relevant information and the technical work related to the models considered in each chapter as well as an electronic version of the thesis in Word and pdf

study that draws on various financial market applications and in particular the Monopoly Dealer simulation developed by Larry Harris[13]. The case study model is developed using EM and VR technology. The chapter proposes an EM – VR merge for a broader foundation of computing tackling the wider agenda for computing in finance.

Chapter 7 considers modelling interpersonal communication in software system development for financial engineering. This is motivated by the need to consider the wider view of the financial engineering activity undertaken within a social network of academics or practitioners. The case study considered is the spreadsheet affine interest rate model developed by Nick Webber. An extension of Webber's model has been developed using Distributed Empirical Modelling (DEM). This model can be interpreted in two ways: (1) as a proposal for distributing the spreadsheet financial engineering activity to suit a group learning environment and a social network of practitioners that are using a financial instrument model for various purposes; and (2) as exposing the need for, and requirements of, computer mediated interpersonal interaction for group financial engineering activity.

Chapter 8 studies SSD in relation to the finance research development cycle (FRDC), with reference to a practical academic case study. It proposes a situated account rather than an abstract account of the finance research development cycle (FRDC). The motivating case study for this chapter is a long horizon event study using UK financial data [Ho00, AH01] undertaken by Keng-Yu Ho[14]. The review of the technical computer support activity undertaken by the author in this case study enriches the understanding of the relationship between SSD and the FRDC and informs a brief analysis of the future prospects of EM technology for data intensive applications.

Chapter 9 concludes with a summary of the original contributions of this thesis to the fields of computer science and finance and endorses Sun's claim [Sun99] that software system development is most appropriately viewed and conducted as a context dependent social activity. The conclusion motivates further research and development activities for EM technology for finance, with the social and human aspects as central considerations in the technical analysis. Further research on various case studies considered in chapters 5 through 8 are suggested and an EM – VR merge is proposed.

---

format.

[13] http://lharris.usc.edu/trading/DealerGame\Default.htm

1.3 Research Contribution

The original specific contributions in this thesis are briefly reviewed below with reference to the relevant chapter.

Chapter 2 identifies a wider agenda for computing in finance that motivates a paradigm shift at the computational level. Currently there is a wide gap in the literature covering computer-based application in finance. Attempts at bridging this gap are found in the literature on AI and its application to finance (see [Deb98, FK95, TL96, Deb94, Pro91]) and sparser literature describing the application of IT at the operational level in the financial enterprise (see [Ess93]), the implementation of trading strategies (see [Gol88]), and various statistical tools used in financial analysis (see [Vin00]). The novelty of the literature review presented in chapter 2 is in considering the wider need of the application of computer-based technology in finance at the three levels (enterprise, market, and investment).

Chapter 3 introduces EM with reference to simple but original case studies: Ordinary Least Square (OLS) regression, the capital asset pricing model (CAPM), the story of a retail trade in New York Stock Exchange, and a distributed stock market game. The OLS regression is implemented as a built-in feature in Excel and the CAPM model is implemented using an Excel spreadsheet by Peter Corvey (another similar implementation of the CAPM is in software accompanying the textbook by Elton (et al, 1995)). The novelty in the implementation of the OLS and CAPM using EM tools is in illustrating the potential of EM technology in generalizing the spreadsheet concept in three respects: presentation, underlying data type, and agency. Describing a retail trade process in the financial market using a special purpose notation such as the LSD notation that can serve in developing an interactive situation model for exploring this trading process is an innovation in modelling the financial trading process. The novelty is in combining the use of descriptive and interactive computer-based media to serve in experiential knowledge construction in connection with the financial trading process. The originality in considering the distributed stock market game is in leveraging enhanced communication by sharing definitive scripts.

Chapter 4 overviews the application of EM technology in various application domains with particular emphasis on establishing a closer integration between the software system development activity and diverse real world activities that are not inherently computer-based. A review of the use of EM technology in various application domains was presented in [Sun99] and [Car99]. The added value of the review conducted in this chapter is

[14] a doctoral researcher at Warwick Business School

in the focus on the theme of establishing a closer integration between the computational and the real world activity. This theme is quite appropriate when applied in meeting the wider agenda of computing in finance, and reveals the importance of investigating the applications and prospect of EM in finance.

Chapter 5 illustrates the novel use of the EM approach in integrating the social and the technical in the financial enterprise with reference to software integration and virtual collaboration. The situated integration model (SIM) presented in this chapter is an original approach to software integration that exploits the more holistic view of the integration activity that EM affords. The EM approach to software integration was the subject of a joint paper with Meurig Beynon that appeared in the Fourth World Conference On Integrated Design and Process Technology, Incorporating IEEE International Conference on Systems Integration (1999) [BM99][15].

The application of EM principles to building virtual environments for collaboration was also the subject of a joint paper with Meurig Beynon that appeared in the Fifth World Conference On Integrated Design and Process Technology [BM00]. The use of EM principles in combination with the concept of Human Information Behaviour described in [Son99] in terms of context, situation and social network is an original approach for building virtual environments for collaboration proposed by the author.

Chapter 6 proposes the original idea of using an Open Financial Market Model (OFMM) for studying the shift from the old to the new trading model. This chapter features two original extensions of the Monopoly Dealer Simulation[16], using EM and VR technology with the aspiration to develop an OFMM. The application of EM and VR technology to financial trading appeared in the Human Computer Interaction International Conference proceeding (2001) [MBG01] and was the topic of a seminar I presented at Laboratoire de Robotique De Paris in France in April 2001.

The proposal of an EM-VR merge to establish a suitable framework for deploying EM and VR technology in modelling social applications in domains such as finance is an original idea of the author to be pursued in future research.

In ***chapter 7***, the original contribution is in re-engineering the affine interest rate model implemented on a spreadsheet by Nick Webber to take into account distributed modelling and computer-mediated interpersonal interaction at classroom level. The idea of viewing computer-based support for financial engineering as a social activity undertaken in a

[15] We have been invited to submit an extended version of this paper to the Journal of the Society of Integrated Design and Process Technology.

network of practitioners or academics is an original idea that was motivated in other literature such as [JES00].

Chapter 8 describes the Financial Research Development Cycle and its associated computer-based support in a novel way with reference to long-horizon event studies conducted by Keng-Yu Ho on financial data from the UK and USA markets. This financial study is pioneering, where research on UK market data is concerned, in its novelty and the wide ranging scope of its quantitative analysis [Ho00, AH01]. The author's novel contribution is in considering the needs and requirements of a closer integration of the software system development activity and the financial research development cycle (FRDC).

In general, my research contributes to two fields of study, computing and finance, which are becoming increasingly dependent on each other, with greater dependence of finance on the computational power and the evolution of communication technology. The combination of these two fields of study is sometimes referred to as computational finance. However, the use of this term is currently limited to the computer implementation of financial and econometric models using fourth generation languages and artificial intelligence tools.

Real world domains and computer-based environments are implicitly and explicitly linked through high-level concepts and low level techniques. Explicit links are supported by commonly available technologies, but implicit links[17] can only be explored and understood with stronger foundations for advanced human-centred technology. Activities conducted in the real world finance domain and in computer-based environments are linked through implicit high level shared concepts, and explicit low level common techniques. Current technologies are primarily aimed at supporting explicit links. Examples of this include encoding trading strategies, automating trading processes, managing information, etc. Mathematical abstraction, statistical analysis, and methodological approaches are favoured in establishing the explicit links. Implicit links are not always easy to detect and are difficult to support as they relate to ontological issues pertaining to the two domains (finance and computing).

The difficulty of identifying the implicit links between finance and computing is exacerbated by the historical legacies of the two disciplines, and the very radical transformations to which they have been subjected since they first came together.

These issues are not easily formulated in either domain. In earlier days, it was easy to define a computer environment as an electronic medium for processing information from input to

[16] Originally developed by Larry Harris

output. A single book on the computer science discipline could cover most of the issues relating to input-process-output activity, its requirements and techniques. With the computer and communication revolution and the wide penetration of internet technology into almost every daily activity, new uses for computer-based environments have emerged both incidentally, and driven by an incentive or strong need. Similarly, the financial industry is witnessing major structural changes that are re-defining the role of market participants and re-shaping core activities in the market. Implicit links are emerging as a cause and effect of change and the need to cope with change. Change is reshaping the boundary of different domains of study and the boundaries of divisions within the same domain. Technologies that are better positioned at explaining and coping with change can play a pioneering role in reinforcing emerging implicit links across different disciplines.

This thesis makes it plausible that there can be a very different relationship between computing and finance, that can be established through interdisciplinary collaboration that exposes and re-enforces the implicit, high level conceptual links and the explicit low level technical links that are respectively associated with the strategic and the technical demands for computing in finance to be identified in chapter 2 (cf. Figure 1.1). Specifically, the thesis reveals that novel software engineering practices, experience based modelling, and alternative programming paradigms in EM can impact on the finance domain at the market, enterprise and investment levels in both technical and strategic respects. In particular, the thesis shows that key concepts in EM (observables, agency, dependency), are well suited to framing key concerns of the application of IT in finance, and that a computer-based modelling paradigm based on these concepts is potentially very well oriented towards satisfying current and emerging demands in finance.

[17] By the term “implicit links” we refer to a common shift in perspective on change that is independently motivated by needs and developments in finance and in computing.

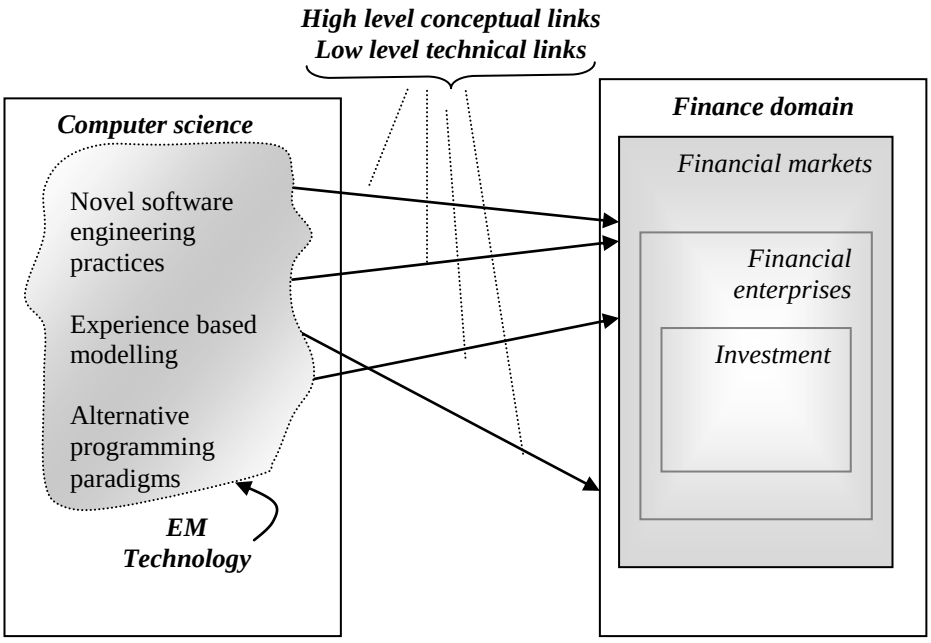


Figure 1.1 Emerging links across the two fields computer science and finance

The specific topic to which the thesis contributes understanding is "***Software System Development and Software Engineering in the Finance Context***". This thesis consolidates on previous EM research on software system development and software engineering [Sun99, BCSW99, SB98, Sun98]. Sun (1999) examined fundamental issues of software system development relating to its essential character, real-world context, human factors, and social factors. His findings were based on EM principles and foundations and supported by a deep investigation of distributed Empirical Modelling in software system development in both theoretical and practical aspects. My thesis builds upon the work of Sun (1999) by considering in greater depth his findings in SSD. It examines the association between Software System Development and its context, and the importance of the situated nature of software system development. The particular context of finance is considered. Key issues concerning the application of computer-based technology in finance motivate a re-thinking of SSD in the finance context through the adoption of EM principles and foundations. Four case studies pertaining to the three identified levels of finance are considered. These case studies support my thesis contribution by:

- establishing new foundations for software system development in the financial enterprise that guide software integration and building virtual environments for collaboration;

- proposing an open financial market model to support software system development meeting various needs of the new financial market model including learning, decision support, and modelling of the trading process;
- considering the importance of distributed modelling for software system development for financial engineering;
- considering the relationship between the software system development cycle and the finance research development cycle.

The thesis sheds light on the development of EM technology from a technical point of view and with reference to its future prospects. With its wider view of the computational activity, EM technology has offered both strategic and tactical support to my research, and empowered it with principles that helped in uncovering hidden relationships between the two domains computing and finance.

Chapter 2

Computer-based Technology In the Finance Domain

2.0 Overview

This chapter introduces the key issues for the application of computer-based technology in finance. It reviews the challenges faced and concludes with the need for a paradigm shift at the computational level to meet the wider agenda for computing in finance.

Section 2.1 considers key issues for the application of computer-based technology in finance at the institutional (integration), market (support of the shift from old to new trading model), and investment levels (support of the finance research development cycle and the group financial engineering activity). Section 2.2 highlights technical and strategic demands for a wider agenda for computing capable of addressing the key issues addressed in section 2.1. Challenges facing prevalent tools and technologies in delivering solutions to problems in the finance domain are overviewed. Section 2.3 concludes with the need for a paradigm shift in computing to address the key issues of the application of computer-based technology in finance at the institutional, market, and investment levels.

2.1 The Key Issues of the Application of Computer-based Technology in Finance

Computer-based technology is becoming an integral part of almost every industry including the financial one. An appropriate use of computer-based technology is a critical success factor at the micro and macro economic levels. The uses and applications of computer-based technology vary across and within industries. This motivates the investigation of the application of computer-based technology in each and every industry and the focus on its diverse applications within a particular industry.

This thesis focuses on the uses and applications of computer-based technology in the finance domain. The key issues of the application of computer-based technology in finance are considered at the institutional (in the financial enterprise), market, and investment levels.

2.1.1 Application of Computer-based Technology in the Financial Enterprise

> *A financial enterprise is mainly a financial intermediary that facilitates the transfer of funds from entities in financial surplus to entities in financial deficit. Financial intermediaries include brokerage firms, banks, stock exchanges, and financial institutions offering trading clearing, or investment services. Appendix 2.1 overviews the organisational structure of typical financial intermediaries.*

The key issues of the application of computer-based technology in the financial enterprise, identified in this thesis, are centred around *integration*. All aspects of integration are needed to face competition in a global market, namely software integration, devices integration, the integration of human and computing activities, and the coherent integration of computer mediated group social activities.

2.1.1.1 Software Integration

The software industry has seen tremendous progress over the last decade. A diversity of tools serving various needs in different industries have been produced. Tools developed for the financial industry are numerous. Some are special purpose financial tools, while others are general purpose tools used in the broad business sector. Although different tools exist separately, success in the financial industry relies on an integrated suite of tools. To meet this requirement, some service providers offer a suite of tools satisfying various needs in the

financial enterprise, in the financial market, and for investment. However, software integration needs remain unfulfilled with a wide range of available tools having different underlying technologies and diverse functionality.

It is difficult to make a proper classification of tools used in the financial industry. Many general purpose business tools are suited for tasks in the financial enterprise. Bocij (et al, 1999) divides business tools into two broad categories: tools that support an organisation's business activities (process control, transaction processing, communication, and productivity), and tools that support managerial decision making. Financial tools form a subset of general business tools. The classification of business tools by Bocij (et al, 1999) is too general, and a more specific classification pertaining to the finance industry is suggested by the author. The latter classification highlights special purpose tools that can be used only in a financial context. Such tools can be divided into three broad categories (cf. Figure 2.1):

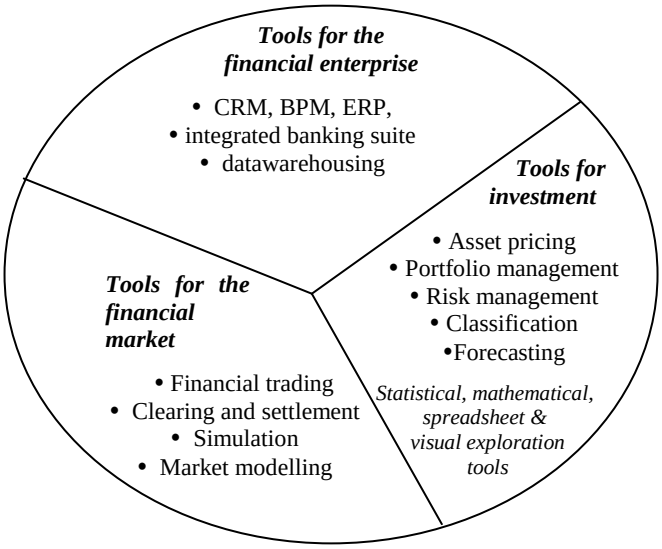


Figure 2.1 classification of tools used in the financial industry

Financial enterprises use different tools to conduct their daily operations and maintain a competitive position in the global market. Integrated suites of tools have been developed by some service providers to offer a unified functionality. There is no clear academic description of the features of tools supporting operations in the financial enterprise. The author referred to the description of commercial computer packages[1] for financial operation as well as non

[1] Extended enterprise ebanking engine (http://www.systemaccess.com)
Integrated back office for corporate & universal banking (http://www.midas-kapiti.com/)

academic journals[2] to provide the description of tools in this category. Appendix 2.2 provides a detailed overview of the features of tools for the financial enterprise outlined below.

- *Enterprise Resource Planning tools(ERP):* handle the internal operations of a financial enterprise firm and automate its processes.
- *Customer Relationship Management[3] tools:* capture all the relevant information about customers and their needs via multiple channels (e.g. telephone, fax, internet), store this information in databases, and analyse it using data-mining and business intelligence tools.
- *Electronic Document Management (EDM) tools:* help in storing, retrieving and managing the workflow of documents across the enterprise, and in undertaking collaborative work.
- *Multi-channel integration tools:* are used to maintain consistency across multiple delivery channels.
- *Business Process Modelling tools:* support the re-engineering of the financial enterprise. Today organisations are facing continuous change, and the effect of this change on business efficiency is highly unpredictable and risky. Modelling and simulation tools have been developed to take the risk out of change.
- *Data warehousing tools:* support the management of the data-warehouse[4]. This involves the process of integrating enterprise wide corporate data into a single repository supporting a variety of decision analysis functions as well as strategic operational functions.
- *e-banking integrated suite of tools:* support online trading and banking services including orders execution, viewing accounts, portfolio monitoring, applications filling, and posting inquiries.

The above overview of tools highlights diverse functionality, widespread application, various needs, and novel uses of the computer. An integrated chain of tools with a common

Wireless financial trading (http://www.jiway.com/)
e-processing for financial services (http://www.sungard.com/)
global financial crossing (http://www.ixnet.com/homepage.html)
Thomson Financial – providers of data, analysis and financial tools (http://www.thomsonfinancial.com/)
IT system for a screen-based order-driven exchange (http://www.isma.org/coredeal1.html)

[2] Banking Technology magazine, Straight Through Processing magazine (Financial Publishing International), Exchange (The magazine of London Stock Exchange), Computers and Finance magazine.

[3] Customer relationship management is referred to as the ability to capture a customer and to satisfy all their needs and requirements with minimum cost and high efficiency

[4] A data warehouse contains a collection of data from various operational systems and sources. It can be used as an integrated information base for making decisions and solving problems.

technology base is necessary to enable the process of turning information into knowledge shared across the financial enterprise.

Appendix 2.1 depicts the organizational structure of different financial intermediaries including banks, stock exchanges, and clearing and settlement institutions. Different departments in the financial enterprise rely on different tools, but a common knowledge base relying on the integration of different tools and a unified data-warehouse is necessary to support top management decisions and inter-department internal communication.

2.1.1.2 The Integration of Human and Computing Activities

With the increased reliance on the power of the computer to conduct the daily operation in the financial enterprise and to support essential change, the integration of human and computing activities becomes more and more important. Software tools and devices are operated by humans and the failure to integrate the human factor in the software system development process reduces the usability and reliability of the tools to perform the intended task and to deal with singular situations as and when they arise. The integration of human and computing activities is necessary at many levels: interface, system analysis, system design and product design. A critical success factor for tools used in the financial enterprise is taking into account the human factor in every stage of the analysis, design, development, testing, and use of the tool. The need for the integration of human and computing activities is highlighted with reference to the above overviewed tools:

- Customer relationship management is a human centric activity and the use of technology to support it requires a high level of integration of the human and device[5] activities. The challenge of computer-based technology is to support the saying of Andrew Fisher "The customer is always right" by finding out more about customers, their buying habits, tastes and spending levels. This is also supported by the view of Mattu (2001) on understanding customer relationship management.
- Business process modelling is not only about processes but also about the human role in participating in the business process model. Technology support for business process modelling should account for the role, behaviour, and interaction of different human participants in the process. This is supported by Davenport's (1996) view published in his article "Why Re-engineering Failed: The Fad that Forgot People" on the cause of the failure of re-engineering projects in the United States. Vidgen et al. (1994) points out that

[5] The term device encompasses the computer and its multimedia support.

BPR has more of an organizational focus than a technical one and that the central tenet of BPR is to devise new ways of organizing tasks, organizing people and making use of IT so that the resulting processes better support the goals of the organization.

- The data warehouse is the information asset of the financial enterprise. Management of the data warehouse requires the integration of the human expertise and the domain knowledge in meta-modelling of the data and its storage and retrieval from different repositories. Tools for the data warehouse should enable human input and feedback at every stage of the information loading. This can transform the data warehouse into a reliable knowledge base for strategic decision support. This is supported by the professional view of Peter Block's *Flawless Consulting* on warehouse projects[6] *"Warehouse projects will fail if the builders get specs from the users, go off for 6 months, and then come back with the 'finished' project. Warehouses are iterative! (I think the word iterative means there are lots of mistakes in the projects.) Builders and users working with each other will not reduce the number of iterations, but it will reduce the size of them."*
- Electronic document management is not limited to the transfer of documents throughout corporate networks. Documents contain valuable information for the enterprise. Failing to integrate the human interpretation of the document content with its permissible workflow reduces the reliability of electronic document management tools. Information security, and human understanding of the document content should be freely introduced at all stages of the electronic document management process to approve, or re-direct, the workflow of documents as and when necessary. The importance of the integration of the human and computing activities related to electronic document management is supported by Waldron's (2000) view on the complexity of the electronic record management: *"The value of electronic records management ERM has suddenly grown – as organisations increasingly hold vital transactions and documents in electronic form, and companies move to electronic processes …. The management of electronic records is complex as it requires a large range of functionality to be implemented well. ERM software may consist of a specialist package, a number of integrated packages, custom-designed software or some combination of these; in all cases, there will be a need for complementary manual procedures and management policies"*

[6] In "Actions for datawarehouse success" at http://www.dwinfocenter.org/success.html

- Electronic banking is becoming the new popular banking model. Face to face interaction of client / bank personnel is replaced by fill-in forms that capture to some extent the client needs and requirements. Multimedia support attempts to replace / complement the human intervention in the electronic banking activity. A proper technology integration and multimedia support are not enough for the success of electronic banking activity. Human judgement and pro-active intervention to deal with singular situations should be enabled at every stage of the electronic banking activity. In discussing the right infrastructure for e-business, Christopherson (2000) points out the importance of gaining a thorough understanding of the business and how well all the core functions interact internally with employees and externally with supplier and customers. He added that true e-business performance can only be achieved if all elements of the infrastructure are integrated, including key business processes and end-user functions.

2.1.1.3 Coherent integration of computer mediated group social activities

Efficient internal (within the financial enterprises) and external (across different financial enterprise) communication is a critical success factor in a competitive and global environment. Generic network tools, telephone, and fax communication tend to be replaced by more sophisticated environments for virtual collaboration. Essinger (1993) identified two roles for computer-based technology in solving the problem of maximizing the effectiveness of communication in retail financial institutions by enhancing the effectiveness of communication facilities used by customers and communication facilities relating to administration. Communication facilities providing a collaborative mode of interaction would better support computer mediated group activities at the administrative and operational level.

Taking account of the situatedness of the collaboration activity within a social network is an important factor. Human interaction with programmable components and devices is shaped by the context of the collaboration network [Son99].

Assessing the collaboration needs and requirements within and across financial enterprises is a challenging task. The primitive description of the organizational structure of different types of financial intermediaries inspires different modes and requirements for collaboration in each and every institution depending on its targeted goals and objectives. The workflow for clearing and settlement of the financial trade transaction (cf. Appendix 2.1) is a direct example highlighting the need for virtual collaboration across and within the financial enterprise. The organizational structure of a bank and a stock exchange (cf. Appendix 2.1) reveals different needs for collaboration. The regulatory body in the stock exchange should

have a high control over the capital raising and trading services activity. Retail banking services, corporate banking, local and global investment benefit from suitable modes and means for collaboration to assess the bank performance and inform top management of banking business progress. Research on the organizational structure of financial enterprises feeds the requirement engineering of building virtual environments for collaboration across and within the financial enterprise. This is an ambitious project that demands a highly interdisciplinary research.

2.1.2 Application of Computer-based Technology in financial markets

> *"The financial markets are simply a huge clearing house where the different financial needs of individuals, companies and governments can be brought together and matched through appropriate pricing mechanisms. They might be actual places or they might be networks of computer screens and telephones. Either way, they address two fundamental needs: what is variously known as saving, lending or investing. The players in the financial markets and in the wider economy can be classified into four broad groups: investors (who have money to spare on assets), companies (which want to borrow money to expand their business), financial institutions (which act as intermediaries between borrowers and lenders), and governments (which act as both borrowers and lenders and play an important role in regulating the market)."* [Vai96]

In the financial market, the uses and applications of computer-based technology are diverse and rising with the shift from an old to a new trading model. Tools for the financial market are classified as educational, operational, and research analysis tools (cf. Appendix 2.2). Relevant academic financial research that investigates the new trading model relates to studies in financial market microstructure[7].

[7] Market microstructure is the academic name for the branch of financial economics that investigates trading and the organization of securities markets. This important field of study has grown substantially since the Stock Market Crash of 1987 [Har98]. Market microstructure analyses security trading and pricing at the institutional/market level. It focuses on modelling the influence of transactions costs on security pricing. By taking a broader view of the financial market structure, this field of study attempts to improve on the 'black box' modelling of equilibrium asset pricing, by accounting for strategic behaviour among trading parties [Abh00-1]. Financial market microstructure studies market structures and market efficiency, transparency, liquidity, volatility, competition, fragmentation, and economic

The key issue for the application of computer-based technology in the financial market is *supporting the shift towards a new trading model*. This involves the development of *computer-based models* reflecting stock exchange integration, enhanced operation (straight through processing, cross border trading) / trading cost reduction, re-engineering, advanced financial market analysis, domain knowledge construction and decision making needs and requirements. This would ultimately serve the purpose of building virtual environments for financial trading adapting to the new trading model.

2.1.2.1 Modelling stock exchange integration

Today, traditional regulated stock exchanges are no longer the only players in the financial market. Traditional exchanges such as NYSE, LSE, and Tokyo stock exchanges, etc., that have a physical location are facing great competition from their electronic counterparts such as NASDAQ, EASDAQ, the quasi exchanges (known as Electronic Communication Networks – ECN), and the alternative trading systems. Alternative Trading Systems (ATS) are growing faster in the US market than in the European markets. ATSs in the US captured 3.1%, 8.1%, and 13% of the total US trading volume in 1990, 1995, and 1997 respectively, and are being used by more than 82% of US fund managers [Hay00].

In face of the competition, traditional exchanges are looking for integration and merger to survive in the global marketplace. Moreover, the old trading model adopted by traditional exchanges is no longer adequate and new trading models are being introduced, revolutionising old execution, clearing and settlement processes. The future of exchanges is unpredictable: trading is becoming easy but the great concern is also to make the settlement and clearing easy as well. The co-operation of exchanges is important but no one can foresee the future trend in co-operation between exchanges.

The technical challenges for integration can be framed in having a unified data structure, communication protocol, type of processing (real time vs. batch), and trading model. However, different trading platforms are difficult to integrate due to system design rigidity and lack of coherent structure. Different data communication standards have been introduced

---

welfare. It investigates orders and order properties, attempts to answer the question "why do people trade?", analyses the type of trading (informed/uninformed), and the behaviour of market participants (investors, brokers, dealers). Appendix 2.3 introduces some basic concepts and definitions in the financial market microstructure area.

(such as FIX[8], XML[9] - extended markup language, OFX[10] - open financial exchange) to facilitate integration purposes, but this integration cannot succeed without a supporting regulatory framework.

Stock exchanges integration entails social, legislative, and technical considerations. It is difficult to study the requirements of social, technical, and legislative integration of stock exchanges separately. Successful integration relies on a consolidated integration plan where technology plays an important role in the requirement engineering of social and legislative integration requirements, and enabling technical integration. The characteristics of financial markets microstructure (cf. Appendix 2.3) reveal the challenges facing the integration of stock exchanges with heterogeneous microstructure (trading sessions, execution systems, and information systems). The agenda for stock exchange integration relies on interdisciplinary research in law, finance, and IT. This research would support an open financial market model informing, amongst other things, the requirements engineering for technology integration of different stock exchanges.

2.1.2.2 Modelling for enhanced operation / lower transaction cost

Online investors are looking for the best trading prices, lowest trading cost, maximum market liquidity, anonymous trading, and easier cross border trading. Haynes (2000) subdivided trading cost into a visible and hidden part (cf. Appendix 2.3). The visible part includes taxes, commission, and spread, while the hidden part of the trading cost includes market impact and delay. In cross-border trading, settlement cost represents a proportionately bigger component

[8] The Financial Information eXchange (FIX) protocol is a messaging standard developed specifically for the real-time electronic exchange of securities transactions. FIX is a public-domain specification owned and maintained by FIX Protocol, Ltd. The mission of the organization is to improve the global trading process by defining, managing, and promoting an open protocol for real-time, electronic communication between industry participants, while complementing industry standards. (For more information visit http://www.siliconsummit.com/fixcenter.html)

[9] The Extensible Markup Language (XML) introduced by the World Wide Web Consortium in 1996 provides a standard approach for describing, capturing, processing, and publishing information over the internet [McG98]. Whereas HTML allows the structural markup of web documents, distinguishing the elements of a page with tags and declaring the physical relationships among the various document elements (new paragraph, head, body), thus allowing human to read it and use it. XML provides a semantic markup (identifying what each particular element means) which is a machine readable structured content.

[10] Open Financial Exchange is a unified specification for the electronic exchange of financial data between financial institutions, business and consumers via the Internet. Created by CheckFree, Intuit and Microsoft in early 1997, Open Financial Exchange supports a wide range of financial activities including consumer and small business banking; consumer and small business bill payment; bill presentment and investments, including stocks, bonds and mutual funds.

of the total trading cost. Efforts are conducted at the international markets level to reduce trading cost by reducing the trading process time and stages. The investment trading process has different stages: order execution, clearance and settlement that can be accomplished in different time intervals up to a maximum of 5 time intervals [Mil99]. This is illustrated in Appendix 2.3.

Straight through processing is the ultimate objective of the new trading model[11]. The seven stages for the execution of an investment trade (trade execution, notice of execution, trade allocation, trade comparison, trade confirmation and reporting, bank notification, clearance and settlement), identified by Milne (1999), are usually executed during five different time intervals. Straight through processing allows the execution of all the stages in one time interval, thus reducing the risk and cost of the trade settlement and improving the market liquidity. Reichardt (1999) argues that the main challenges facing the application of straight through processing is the need for large investment in real-time technology, a deep understanding of the related business issues, industry co-operation on common issues, the acceptance of standards, and the integration of front-office and back-office technologies.

A major challenge facing global electronic trading is providing straight through processing (STP) of transactions, re-designing workflow and processes, and determining the data protocols and connectivity.

Facilitating cross border trading is another challenge facing stock exchanges seeking survival in the global market. Cross border trading requires a geographical spread of the technological, social and legislative infrastructure of a stock exchange. Extending the technology infrastructure of financial trading requires an appropriate re-engineering of the trading system to accommodate different legislative and social framework. This will be reflected in the interface, internal design, and networking of electronic trading systems.

2.1.2.3 Modelling the new financial trading environment

Computerisation is about to overtake markets that traditionally depended on physical presence to bring buyers and sellers together in one place. A new trading model is emerging where individual investors have access to a wealth of market data, research can be easily conducted, plenty of trading choices are offered, online brokers have a retail focus, and a new global trading community is formed [Per99]. The new financial trading model demands new

[11] Reichardt (1999) defined Straight Through Processing (STP) as “fully automated, hands-free processing of security transactions from the fund manager’s decision right through to settlement, reconciliation and reporting”. It is also defined as “the vertical integration of services from trading to settlement reconciliation, or processing without manual intervention”.

legislation for regulating exchanges, quasi-exchanges, and global financial markets. This new legislation should be concerned with taxation, IT systems, operational risk, new standards (transparency, record keeping, listing), legal identity to quasi-exchanges, reduction of national market monopolies, and delivery of financial services over the internet.

Attempts at modelling the new trading environment are approached in: i. academic financial research; ii. interdisciplinary research in computer-based technology and finance; and iii. research on novel software engineering practices for the financial markets.

i. Modelling the new financial trading environment, in academic financial research, involves models for testing market efficiency (cf. Appendix 2.3) and analysing the microstructure of financial markets. Typical research in this area includes, among others, the work undertaken by Oedan et al (2000) - analysing the behaviour and profitability of online investors; Rime et al (2000) - analysing the impact of the introduction of electronic brokers on the structure of foreign exchange market (FX); Jennings et al (2000) - comparing floor trading in NYSE to over-the-counter trading in NASDAQ with reference to best execution prices[12]; Coval et al (2000) - analysing the information content of the ambient noise level in the Chicago Board of Trade's 30-year Treasury Bond futures trading pit; Rolls (1984) - inferring the effective bid-ask spread from security returns (the first-order serial covariance of rates of return); Harris (1988), Hsia, Fuller, and Kao (1994) - extending Roll's model to solve the problem of imaginary numbers in the estimation of the effective bid-ask spread; Eckbo and Smith (1998) - monitoring the movement of insider trading in technically advanced financial markets; Harris (1998) – presenting a detailed overview of how existing markets are organised and regulated, and attempting to answer the question of what are the markets supposed to do and for whom; and Taylor et al (1998) - assessing the relationship between a trading system operated by a stock exchange (SETS operated by London Stock Exchange) and the trading behaviour of heterogeneous investors who use the exchange. In the latter model, Taylor used the cost-of-carry model of futures prices to estimate the transaction costs and trade speeds faced by arbitrageurs who take advantage of mis-pricing of FTSE100 futures contracts relative to the spot prices of the stocks that make up the FTSE100 stock index.

ii. Inter-disciplinary research (computer and finance) can help in the analysis of the effect of structural change in financial markets. The study undertaken by Konana, Menon, and

[12] Best execution is assessed in terms of trade prices and contemporaneous quoted prices, speed of execution, and liquidity enhancement.

Balasubramanian (2000) evaluates the impact of online investment on the efficiency[13] of financial markets and the trading system in the short and long term horizon. Two models describing the investor choice process and the revenue flow and incurred cost in online trading are presented. The models take the form of static diagram illustrating the workflow of a financial transaction.

iii. The shift from an old to a new trading model motivates the re-engineering or the development of new virtual environment for financial trading and the adoption of new software engineering practices. So far, there is no proper description of the new trading model that can inform the re-engineering / development of new trading systems. Knowledge of the new trading model is gradually constructed and continuously evolving over time to accommodate new patterns and modes of trading and behaviour of different market participants. This poses a challenge for computational models of software system development that begins with an engineering requirement stage and terminates with an end product delivery at a final stage. The weakly structured knowledge surrounding the new trading model motivates the re-thinking of the software engineering and software system development process to take into account a growing boundary for the system being developed and continuous knowledge construction informing improvement of the final end-product delivery. Providing appropriate interfaces for virtual environments for financial trading is another major challenge. In the context of financial trading, the behaviour of different trading parties (investors, brokers and dealers), trading signals, economic and financial indicators, and trading systems, constitutes a complex environment which is difficult to capture in a mathematical model, a computer simulation, or a textual description. This motivates the integration of different technologies to support the development of virtual environments for financial trading meeting the requirements of the new trading model.

2.1.3 Application of Computer-based Technology in Investment

"Investment can be divided, both in theory and practice, into two parts security analysis and portfolio management. Security analysis attempts to determine whether an individual security is correctly valued in the marketplace. Portfolio management is the process of combining securities into a portfolio tailored to the

[13] In [KMB00], the processes determining market efficiency include: order flow, price discovery, and order execution. In that context, two types of market efficiency are identified: the efficiency perceived by investors, and the real efficiency of the transactions and the patterns of information flows beyond the interface.

> *investor's preferences and needs, monitoring that portfolio, and evaluating its performance. The three main themes of investments are: the risk-return tradeoff and the principle of modern portfolio theory and efficient diversification, active versus passive portfolio management, and equilibrium pricing relationships" [BKM96].*

Associated with investment is the financial engineering process that consists of developing new financial instruments that meet different risk/return profiles. The elaboration on investment themes and the different types of financial instruments is given in Appendix 2.4.

Tools for investment including financial analysis and modelling tools (General purpose OLAP (Online Analytical Processing) , Data mining, Spreadsheet , Visual exploration tools, Risk assessment tools, Market prediction tools, Portfolio management tools, Numerical Libraries, Statistical packages), educational tools and financial engineering tools are overviewed in Appendix 2.2.

The key issues of the application of computer-based technology in investment are supporting the financial engineering activity as a group social activity whilst providing increased flexibility for the financial modelling activity, and providing greater support for the finance research development cycle that informs investment and financial market analysis.

2.1.3.1 Providing greater support and flexibility for the distributed financial modelling activity

Finance is borrowing new principles from engineering, and the term "financial engineering" refers to the development of new financial instruments meeting clients' requirements to hedge against risk and maintain a profitable position. Like engineers, financial experts build prototypes of a financial model and share it across a network of financial experts to analyse and discuss its prospect in meeting clients' needs and requirements.

Developing new financial instruments satisfying rising needs and requirements in a global market does not involve only financial analysis relying on the implementation of mathematical models. The financial engineering process is a complex collaborative activity that results in the delivery of a final product tailored to customer needs. Providing a computer-based support for the financial engineering activity is highly dependent on effective support for interpersonal interaction in the course of engineering a financial product.

2.1.3.2 Providing greater support for the financial research development activity

Financial research is broadly described in terms of data collection, methodologies selection, methodologies implementation, result analysis, and conclusions. Computer-based support of the financial research development cycle varies at various stages of this activity. It may take the form of data storage and retrieval in the collection stage; spreadsheet or high level language programming support in the implementation and result analysis stage. Various tools exist to support particular stages in the research development cycle. A closer view of the financial research development activity can potentially provide us with means for an adequate software system development support.

2.2 Meeting The Needs of the Wider Agenda of Computing

Section 2.1 overviewed the key issues for the application of computer-based technology in the finance domain at the enterprise, market, and investment levels. They can be summarised as follows: integration that takes different forms; novel computer-based modelling to accommodate change and rising needs; and novel software system development practices. The above identified issues suggest a wider agenda for the computational activity that draws on existing computational practices and solicits a broader foundation of computing capable of addressing technical and strategic demands of the application of computer-based technology in finance.

The *technical demands* refers to the characteristics that we expect of computer-based technology and the concerns that have to do with how computer software is generated. These demands include: i) dealing with experiential issues; ii) enabling closer integration of the manual and automatic interactions; iii) promoting flexible software system development; and iv) supporting new computer-related technologies.

The *strategic demands* are the aspirations that we have for real-world activities using computer-based technology. They entail: i) providing qualitative decision support; ii) facilitating open-ended flexible modelling; iii) integrating the software system development activity with activities in the real world; iv) establishing different paradigms for human-computer co-operation that are better positioned to support business process modelling; and v)

integrating diverse kinds of agency. Meeting these strategic demands is referred to throughout this thesis as “the wider agenda for computing”.

The wider agenda for computing suggests a more intimate (semantic) relationship between the computing activity and activities in the real world (e.g. business and finance related) than that offered by conventional computer science. The latter assumes an application interface that separates the computer-related activities (design, analysis, and programming) from the real world activities and their corresponding domain analysis. This is facilitated by the abstraction and circumscription of the real world applications (cf. Figure 2.2). Figure 2.3 suggests an evolving semantic relationship between the computing activity and activities in the real world. Such a relationship can potentially provide suitable computer-based support for the real world activities by integrating computer-related analysis and modelling with real world domain analysis and modelling.

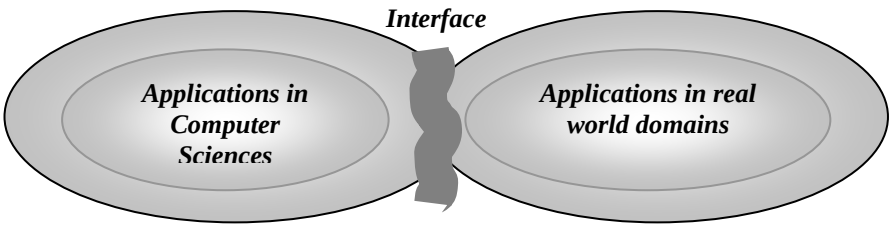


Figure 2.2 The interface between applications in computer sciences and applications in real world domains

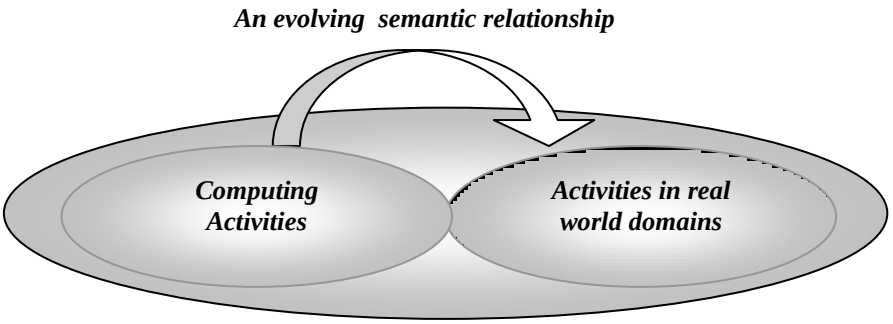


Figure 2.3 The wider agenda for computing: a more intimate relationship between computing and activities in real world domains

Current tools and technologies face major challenges in meeting the technical and strategic demands of the wider agenda for computing. These challenges are overviewed in the following sub-sections.

2.2.1 Tools limitation

The software industry has witnessed a noticeable progress in the development of various tools meeting different requirements and needs. However, many obstacles are posed in exploiting the use of these tools in meeting the need and requirements of a growing and changing financial industry. These include:

- *The adoption of rigid software system development approaches*: A major reason why the benefits of software implementation have been so slow in coming is that system complexity is not adequately managed. Financial enterprises and markets require more complex trading and operational systems with the shift toward integration and enhanced operation (STP, cross border trading, low transaction costs). This makes the software system development activity more complex. Introducing changes to complex applications is very difficult or even impossible in the absence of an interactive flexible approach to software system development that takes account of a growing system boundary and the social aspect of an application domain.
- *The difficulty of moving from the problem domain (real world application) to the solution domain (world of programs and systems):* There is currently a wide gap between research and development, and target goals and current achievement. Bridging these gaps is not only a technical issue but also entails management and communication issues. Heeks (1998) argues that most IT systems failures are due to a conception-reality gap between the rational conceptions of information system initiatives and the behavioural/political realities of organisations. This is also acknowledged by ongoing research that attempts to comprehend the relationship between software engineering and business process modelling [FRKBPJ00].
- *The rising cost and complexity:* Current pressures are obliging the software industry to build faster applications with more control and higher quality at lower cost and effort. However, as complexity increases, cost and development time escalate.
- *The limitation of the underlying technology:* many software products launched in the market are not well-suited for describing interaction in a multi-user environment with many applications. Interaction in the financial market is subtle and complex, and building

trading systems that accommodate this high level of complexity requires a broad foundation of computing borrowing from disparate disciplines.

- *The limited functionality:* most of the tools are built with preconceived functionality in mind. This reduces the adaptability of the tools to rising needs and requirements. Addressing this issue involves placing greater emphasis on the modelling rather than the programming activity. The choice of modelling paradigm also has a great influence on the functionality of the end product. Making an appropriate translation of the model into an application program is also a challenging task. The new trading model is weakly structured. This motivates an open-ended modelling paradigm that continuously informs end product development. This is very ambitious given the difficulty of establishing a semantic relationship between the end product and the continuously evolving model.
- *The limitation of the underlying theory:* most of the financial analysis tools (including financial instruments pricing, risk assessment, neural networks, and investment strategies, etc..) rely on an underlying mathematical model. Investment themes (cf. Appendix 2.4) reveal a striking conflict in adopted theories. This is expressed in the conflict in equilibrium pricing relationships [Rol77] (the use of capital asset pricing model [Sha64, Lin65] vs arbitrage pricing theory [Ros76]) and trading strategies [Gol88] (fundamentalist vs chartist). Moreover, the efficient market hypothesis [Fam70, Fam91] (cf. Appendix 2.3) raises questions concerning the use of tools that analyse past historical data and inform future action in efficient markets where prices follow a random walk [Mal99]. These conflicts motivate the consideration of an experience-based approach to building tools for financial analysis and decision support. Financial modelling tools that implement a formally established financial theory might fail to deliver the intended decision support. This is because financial theories are based on a set of assumptions and take into consideration a limited number of economic and financial factors. This undermines the reliability of financial modelling in supporting proper decision-making, especially when the assumptions do not hold and more economic and financial factors need to be taken into consideration.

2.2.2 Technologies limitation

Underlying technologies used in the development of tools for the financial industry are numerous. They include Object Oriented, Artificial Intelligence, Database, VR, and web technologies. Many limitations are acknowledged in these technologies relating both to foundational principles and to issues for applications and implementations.

2.2.2.1 Object Oriented Technology

Motivations

The object oriented approach is adopted in the development of many tools and systems in the financial industry[14]. This is motivated by three considerations: the possibility of developing applications of increasing complexity; the need to reduce the cost of development and maintenance; and the evolution of classical business applications from storing / retrieving / coding information to representing complex objects and more elaborate processing encompassing business control, rule-based reasoning and decision making [BGV97].

Historical background

Object oriented technology relies on the concept of object introduced in [DN66], the theory of abstract data types formalized in [LZ75], the concept of message passing and inheritance introduced in [HBS73], and the linking of objects by the "is-a" relation introduced in [Win84].

Applications

The widespread use of object technology is attributed to two reasons: objects seem to be a good abstraction of real world entities and can be used in modelling complex systems; and objects are modular and can be re-used in complex applications.

In Object Oriented Modelling, the object concept is associated with encapsulated units of data and operations. Documents, transactions, products, individuals in our daily business life are modelled as objects such as invoices, orders, and customers. In relating to each other, to system users and to the outside world these objects are self-managing. A business object is a representation of a thing active in the business domain, including at least its business name and definition, attributes, behaviour, relationships, rules, policies and constraints. A business object may represent, for example, a person, place, event, business process or concept.

Challenges

Major challenges face object oriented technology and its application in different areas. These include:

- *Design and development problems:* Despite its popularity, object oriented technology cannot solve all design and development problems. Bouzeghoub (et al, 1997) argues that there is still considerable confusion and controversy over key concepts of OO technology, such as encapsulation[15], inheritance[16], and polymorphism[17].

[14] See technology used in Misys International Banking Systems Ltd (http://www.misys-ibs.com/).
[15] Encapsulation of the data is the application of the principle of abstraction into objects and operations that manipulate the object (classes and methods). Objects are accessible only by means of their visible operations, the implementation of objects is hidden from the program that manipulate them).
[16] Inheritance is the essential distinguishing feature of object languages. It can be simple or multiple and can support polymorphism of operations. It is basically the mechanism of transmitting the properties of a class to a subclass.

- *The difficulty of object identification:* Another challenge facing object oriented technology is the difficulty of identifying objects and their behaviour in a system at an early stage. This motivates the support of object oriented technology with other technologies capable of modelling a weakly structured domain and identifying candidate objects and their reliable repetitive patterns of behaviour. Such a supporting technology would precede OO analysis and rely on an experience based approach to understand a problem domain.
- *The OO analysis and design gap:* Reconciling object oriented analysis and design is a major challenge acknowledged in [Kai99].
- *The difficulty of modelling weakly structured domains:* The object oriented approach does not deal with partial knowledge of the real world. Object attributes are predefined at an early stage in program design. Problems arise with the emergence of new knowledge which reshape the real life aspect of these objects. Inheritance and encapsulation minimize the overhead of code re-engineering to integrate the change, however the amount of adjustment is still considerable. Exploiting object oriented software engineering methodologies for business process modelling is challenged by the weakly structured domain knowledge [FRKBCJ00].
- *The minimal added value in certain application domains:* Eber (1999) argues that object oriented technology does not deliver any added value to financial modelling as compared to a high level language such as C++ or Pascal. Object Oriented technology is just another way to write a program and it does not lead to much improvement in financial reasoning or financial model implementation.

2.2.2.2 AI technology

Motivations

Broadly speaking, AI technology is concerned with creating computer programs that can perform actions comparable to human decision making [Shap92].

Historical background

The underlying assumptions of a large body of research in AI rely on Newell and Simon's (1976) physical symbol system[18] hypothesis [NS76] that "a physical symbol system has the necessary and sufficient means for general intelligent actions". The truth of this hypothesis is

[17] Polymorphism is the re-definition of inherited method to give it a different implementation (to code it differently).

[18] A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure) [NS76].

to be assessed by experimentation, and computers provide the perfect medium for this experimentation.

AI techniques are classified into two main classes: methods for representing and using knowledge and methods for conducting heuristic search. These aim at developing adaptive systems (systems that learn or adapt in response to outside events according to their previous 'experience'). Artificial intelligence technologies include neural networks[19], genetic algorithms, fuzzy logic, etc. These techniques have been used for financial forecasting, credit rating, customer profiling and portfolio management [Dav91, FK95, Deb94].

Applications

AI practice has found widespread application in many areas including finance. The application of artificial intelligence in finance [TL96] investigates the use of knowledge-based systems (also called expert systems) to investment decision making. Decision support systems for investment integrate mathematical models, a data source, and a user interface. Knowledge based systems are decision support systems whose database include relevant theory, facts, and human knowledge and expertise. The finance and investment literature often fails to discuss even briefly computer implementations of the economic theories and principles involved in investment management. According to Trippi (et al, 1996), artificial intelligence in finance can be thought of as a bridge between finance and information science.

Artificial neural networks have fuelled the recent interest in non-linearities in financial data. Most econometric methods used in testing financial theories are designed to detect linear structure in financial data. For instance, the Capital Asset Pricing Model (CAPM) and the Arbitrage Pricing Theory (APT) are based on linear models of expected returns. However, many aspects of economic behaviour may not be linear. Experimental evidence and causal introspection suggest that the relationship between an investor's attitude towards risk and his expected return is non-linear. Modelling the strategic interaction among market participants, the process by which information is incorporated into security prices, and the dynamics of economy-wide fluctuations is clearly beyond the scope of linear mathematical models. It seems that modelling of non-linear phenomena defines a natural frontier for financial econometrics [CLM97]. On this basis, Fama (1970) argues that traditional economic theory

[19] Neural networks are a collection of mathematical techniques that can be used for signal processing, forecasting, and clustering. Neural networks can be viewed as non-linear, multi-layered, parallel regression techniques. In simple terms, neural network modelling is like fitting a line, plane or hyperplane through a set of data points. A line, plane or hyperplane can be fitted through any set of data and define relationships that may exist between the input and outputs. There are two classes of neural networks: supervised neural nets (techniques for extracting input-output relationships from data); and unsupervised neural nets (techniques for classifying, organizing, and visualizing large data sets) [Dko98].

cannot be used to devise mechanised methods for predicting market movements at all. Economists of a new generation are creating new models and tools that can capture non-linearities in economic phenomena. Artificial neural networks are an alternative to non-parametric regression and have received recent attention in the engineering and business communities.

The importance of the use of artificial intelligence techniques in financial modelling is emphasised by Ridley (1993). In Ridley's view market trading relies on intuitive and complex reasoning on the part of the human trader. The trader interprets and deciphers many factors surrounding the market of interest. The factors can be wide ranging and can vary over time. This kind of changing structural relationship implies that the form of decision process required by market trading is not open to precise calculation and therefore not open to mechanisation [Fam70].

Challenges

Despite its penetration in various application domains, artificial intelligence technology still faces many challenges:

- *Rigidity of the AI solution:* In facing the challenge of building real-world systems, AI faces a criticism of the rigidity of the solution it produces. Some researchers attribute this fact to the difficulty faced by AI to scale well beyond the relatively small domains to which they have been applied. Moreover, the performance of an AI system is extremely sensitive to the representational choices made by its designer and these are subtle in the face of the inevitable deviations from the norm found in the real world. As was first pointed out by Dreyfus (1979), the success of an AI system appears to be strongly correlated with the degree to which the problem domain can be treated as an abstract micro-world which is disconnected from the real world at large.
- *The difficulty of abstracting various aspects of the complex real world domain:* The fundamental AI approach of subdividing mental activities into independent tasks (ordinary, formal, and expert task), the hypothesis of the physical symbol system, the focus on structured knowledge management, the four-step problem solving mechanism, and the knowledge management techniques, are viewed by many as an oversimplification when solving complex inter-related problems requiring ordinary, formal, and expert tasks. In contrast, many modern approaches to problem solving emphasize the importance of thinking in embodied and embedded terms [Cla01].
- *Dealing with domain specific characteristics:* Despite the developing importance of the use of artificial intelligence in the financial community, the Efficient Market Hypothesis (EMH) still poses a significant challenge for a machine learning AI system. If the EMH

holds true then no solution exists to predict the market, and all AI systems are doomed to fail no matter how sophisticated they are. If capital markets are efficient, then changes in stock prices should be associated exclusively with new information. This implies that information, once available, triggers a rapid process of adjustment for prices to their correct level, where it once more reflects all available information. This in turn implies that the movement in prices is random and is based on future events. Unless an AI system can anticipate the outbreak of war, or political events, or financial news, it is condemned to failure within the EMH framework [Mal99].

The above limitations facing conventional approaches to AI motivate a new foundation for artificial intelligence that emphasizes an experience based approach to understand a domain and construct the corresponding computer-based artefact.

2.2.2.3 Virtual Reality technology

Motivations

Virtual Reality technologies [MT94, LBBRS97, Kru83] supply virtual environments that have key characteristics in common with our physical environment. Viewing and interacting with 3D objects offers greater realism than abstract mathematical and 2D representations of the real world. In that respect virtual reality can potentially serve two objectives: (a) reflecting realism through a closer correspondence with real experience, and (b) extending the power of computer-based technology to better reflect "abstract" experience (interactions concerned with interpretation and manipulation of symbols that have no obvious embodiment e.g. share prices, as contrasted to interaction with physical objects). The main motivation for using VR to achieve objective (a) is cost reduction (e.g. it is cheaper to navigate a virtual environment depicting a physical location such as a theatre, a road, or a market, than to be in the physical location itself), and more scope for flexible interaction (e.g. interacting with a virtual object depicting a car allows more scope for viewing it from different locations and angles). Objective (b) can be better targeted because the available metaphors embrace representations in 3D-space (c.f. visualization of the genome).

Applications

The dominant emphasis in current uses of VR is on the exploration of a real physical object (e.g. car, cube, molecule, etc..) or a physical location (e.g. shop, theatre, house, forest, etc..). In the course of exploration the user is immersed in the VR scene, and can walkthrough or fly through the scene. The user's body and mind integrate with this scene. This frees the intuition, curiosity and intelligence of the user in exploring the state of the scene. In a real context, agents intervene to change the state of current objects/situations (e.g. heat acts as an agent in

expanding metallic objects, a dealer acts as an agent in changing bid/ask quotes and so affects the flow of buyers and sellers).

The use of virtual reality technology in finance was pioneered by the NYSE who developed the first large-scale virtual reality environment for navigating the stock exchange trading floor and the Advanced Trading Floor Operations Center in a 3D virtual scene[20].

The 3-D Trading Floor consolidates the data streams of NYSE operational activity into one highly advanced, three-dimensional graphic visualization system, revolutionizing the way the NYSE monitors and responds to systems and stock-related events. The 3DTF is a completely interactive, virtual representation of the trading floor that enables the NYSE operations staff to pinpoint complex systems activities and stock-related activity with remarkable visual clarity.

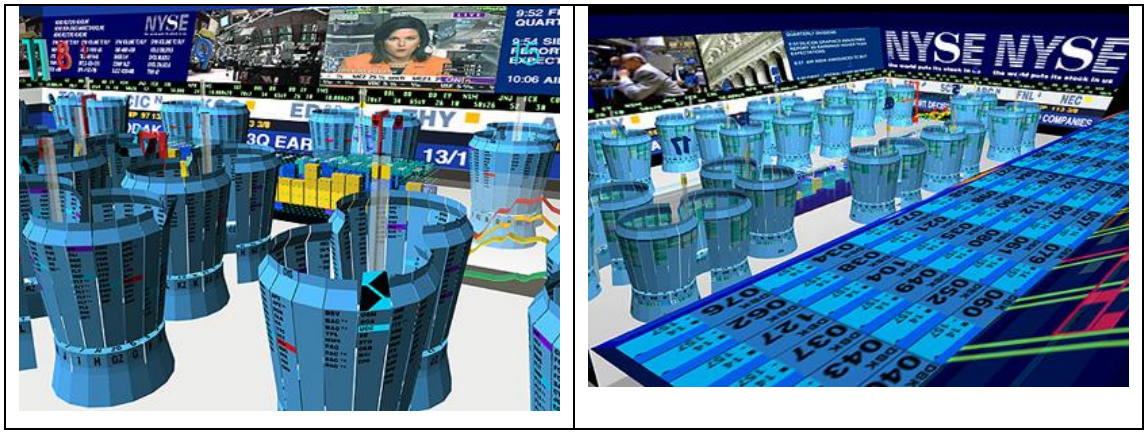


Figure 2.4 The use of VR technology to depict the trading activity in NYSE

Such an application of VR technology in finance attempts to merge the mechanism of a trading process with the analysis of the results of the trading process in a synthesised 3D visual environment. This highlights the importance of the human visual system and its role in supporting knowledge construction about an environment.

Challenges

VR technology faces many challenges. These include technical challenges and limitations in respect of various application domains.

- *The lack of basic foundations for building VR environments:* There is no agreed software development practice for building virtual worlds and scenes. Current immersive virtual environments systems use a wide variety of programming methodologies and scene

[20] http://www.nyse.com/floor/floor.html

description languages (data flow models, visual languages, etc.), and although many support importation of VRML scenes, there is no agreement on how behaviour and interaction should be described [VEa98].

- *Limitations in respect of various application domains:* The challenges facing VR technology include integrating the social aspect of real world environments with their corresponding 3D visualization and animation in the 2D scene [MBG01]. Ontological issues bearing on the questions: "What is reality? What is virtual reality? What is the correspondence between the two?" are motivated in the application of VR technology to different application domains. The challenges facing the use of VR technology in modelling social application domains are due to limitations in the technology and in the principles underlying the technology. The technical limitation is due to the fact that information technology is not mature enough to capture all human sensations and simulate it by devices. Attempts at simulating human cognition and sensations with the aid of multimedia devices is an interdisciplinary research involving electronic devices, information processing systems, cognitive biology, human biology, sociology. VR technology lacks supporting principles for the integration of the social aspect in a virtual environment. Representing the human role in the virtual environment still poses major difficulty. Avatars are still primitive and limited to iconic representation of a human being.

2.2.2.4 Database Technology

Motivations

Management of financial data (storage, retrieval and update) is very important in conducting financial analysis and forecasting. Data management paradigms evolved from files[21] to groupware[22] and databases[23] [Bla01]. Databases are concerned with storing and processing data on computers. Data analysis, data modelling, and data security are key issues addressed

[21] In the file processing approach data is accessed sequentially or randomly through a physical program interface. The file processing approach is not adequate for computer environments including many applications accessing many files.

[22] Groupware is software that manages unstructured information for collaborating users. Groupware organizes data into documents that consists of items. An example of a groupware data management application is Lotus Notes. Groupware has limited support for data management actions such as multi-user access, error recovery and flexible queries.

[23] A database is a permanent self descriptive store of data. It contains the data structure (schema) and the data. A database manager is the software managing access to the database. Different database paradigms exist: relational (where data is perceived as tables, and data is accessed by a relational database management system), multidimensional (tailored for analytical applications with complex relationships between data), object oriented (based on the concept of embodied objects, and relationships between objects established through pointers).

in database design and management. Database systems were motivated by the need to overcome the problems facing file processing applications, where files of structured data are processed by programs. Database systems first began to appear in the 1960s but, since then, have witnessed a major change in concepts and technology. A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access that data [KS86]. Its primary goal is retrieving and storing information into a database. A database system provides the user with an abstract view of the data at three levels: physical level (describe how the data is stored); conceptual level (describe the data and relationship between data stored in the database), view level (describe part of the entire database). Different data models can be adopted at the conceptual and physical level: object-based logical model; record-based logical models; or physical data models.

Applications

Different financial markets rely on different financial data sets. Examples include the CRSP[24] Database for USA financial market data, the LSPD[25] (London Share Price Database) for UK market data, and the S&P's Emerging Markets Data Base (EMDB[26]) for Emerging markets data.

Challenges

Many challenges are facing modern database technologies. These include:

- Providing appropriate visual interfaces
- Enabling selective access to data in world databases: this includes complex retrieval and search query.
- Providing end user programming: despite the advance in visual query languages, the aim of providing end user accessibility to large databases is difficult without resorting to high level 3rd generation languages. Users of World financial databases do not rely solely on the provided visual query to extract data for a single firm. High level programming in Fortran or C is usually resorted to in advanced manipulation of large financial databases.
- representing complex and various forms of data (e.g. relating to multimedia)
- Maintaining and representing complex relationship between data
- representing complex relationships between data of different types
- providing novel style of interaction with the database.

[24] www.crsp.uchicago.edu

[25] http://www.lbs.ac.uk/ifa/Services/London_Share_Price_Database/london_share_price_database.html

2.3 Conclusion

This chapter has identified key issues of the application of computer-based technology in finance at the institutional, market, and investment levels. This motivates the reconstruction of computing in a wider framework capable of addressing strategic and technical demands of computer-based support for the financial domain. The chapter considered the limitations of prevailing tools and technologies in meeting the wider agenda for computing in finance. This suggests a paradigm shift in software system development for the financial enterprise, the financial market, and investment.

Key issues for the application of computer-based technology in the financial enterprise motivate the need for a shift in perspective ***from***:

- methodical ***to*** amethodical software system development;
- formal approaches ***to*** situated experience based approach to software system development
- considering software system development as a technical activity ***to*** considering software system development as a social activity that needs computer-based support
- closed ***to*** open software system development

Key issues for the application of computer-based technology in the financial market motivate the need for a shift in perspective ***from***:

- programming an end product ***to*** modelling complex, weakly-structured knowledge of a domain
- abstract mathematical modelling ***to*** experience based modelling of the financial market
- rigid boundaries solution ***to*** growing boundaries solution
- full automation ***to*** human engagement in semi-automated activity

Key issues for the application of computer-based technology in investment motivate the need for a shift in perspective ***from***:

- a rigid financial research development cycle ***to*** a situated account of the financial development cycle
- closed ***to*** open functionality in financial modelling
- from supporting the financial engineering activity as a personal activity ***to*** supporting financial engineering as a group social activity.

[26] http://www.spglobal.com/indexmainemdb2000.html

Key issues for the applications of computer-based technology in the financial enterprise, financial market, and investment reveal a wider agenda for computing in finance that draws on:

- basic principles and foundations motivating a paradigm shift in software system development and novel computer uses and evolution
- integrated features in disparate technologies
- holistic approaches to tackle domain specific needs

The limitations of current tools and technologies to address the implications of the wider agenda of computing in finance on software system development practices and computer uses motivated the investigation of a suitable framework for deploying current prevalent technologies. EM technology aspires to develop the requirement engineering and to establish principles for such a framework that can potentially address the wider agenda for computing in finance. The application and prospects of this novel technology in the finance domain is the theme tackled in the next chapters. Case studies taken from the finance domain and addressing the key issues for the application of computer-based technology at the three finance levels are considered.

Chapter 3

Empirical Modelling: A New Approach to Computer-Based Modelling

3.0 Overview

Chapter 2 presented an overview of the application of IT in the finance domain. This chapter focuses on a particular technology – Empirical Modelling Technology – developed at the department of Computer Science at Warwick University. Empirical Modelling technology provides a broad computational framework encompassing foundations for software system development, artificial intelligence, computer aided engineering design, business process modelling, and computer aided manufacturing. This puts EM technology in a position to potentially deliver solutions to problems in the finance domain.

Section 3.1 Introduces EM as a suite of key concepts, techniques, notations and tools. These are illustrated with reference to simple examples drawn from the finance domain. Section 3.2 highlights the distinctive qualities of model building in EM framed into: a) the focus on state as experienced; b) the use of artefacts for knowledge construction and c) the use of definitive scripts as a framework for distributed communication.

Section 3.3 concludes with motivating the use of EM to tackle technical and strategic demands for the wider agenda for computing.

3.1 Introduction to EM

«*Empirical Modelling (EM) is an approach to computer-based modelling that has been developed at the University of Warwick* [EM web site]. *It combines* ***agent-oriented modelling*** *with* ***state representation*** *based on* ***scripts of definitions*** *that capture the* ***dependencies*** *between* ***observables****. Unlike conventional modelling methods, its focus is upon using the computer as a* ***physical artefact*** *and modelling* ***instrument*** *to represent* ***speculative and provisional knowledge****. The central concepts behind EM are: definitive (definition-based) representations of state, and* ***agent-oriented analysis*** *and representation of* ***state-transitions****. In broad terms, changes of* ***state*** *within a system are interpreted with reference to a family of observables through which the interactions of* ***agents*** *in the system are mediated.*» [Bey99]

This section introduces Empirical Modelling as a set of principles, techniques, notations, and tools (cf. Table 3.1). It assumes some familiarity with programming paradigms and computer notations.

I. Empirical Modelling key concepts • Observation / observable • Agency / agent • Dependency • State/ definitive representation of state • Definitive script • Agent oriented analysis • Representation of state transition	**II. Empirical Modelling techniques** • Construe a situation • Construct an Interactive Situation Model (ISM) • Metaphorical representation through ISM
III. Empirical Modelling notations LSD account of observables Agent { state oracles handles derivates protocols }	**IV. Empirical Modelling tools** • EDEN interpreter • Distributed variant of EDEN

Table 3.1 Empirical Modelling Framework

3.1.1 Key concepts in EM

Empirical Modelling technology focuses on state representation. Empirical Modelling technology establishes principles that favour state representation over behaviour[1] automation at a preliminary stage of the software development process. Handling state is a major theme in computer programming [Bey98]. In EM, representing state in a comprehensible fashion is addressed in a definitive script that specifies the following information pertaining to state [Yun92]:

- *Observables as constituents of the state:* An observable is a characteristic of a subject to which an identity can be attributed. An observable in EM can be physical or abstract in nature. The clock is an example of a physical observable. The true value of a security in the financial market is an example of an abstract observable.
- *Dependencies between observables:* A dependency represents an empirically established relationship between observables. The attribute "empirically established" reflects the fact that a dependency is not merely a constraint upon observables, but reflects the impact of change in the value of one observable on other observables. Dependencies play a significant role in construing phenomena [Bey99].
- *Agents as instigators of state change:* An agent in EM is an instigator of change to observables and dependencies. An agency is an attributed responsibility for a state change to an observable. A literal dictionary definition [rhyme] of the term agent is "a substance that exerts some force or effect". The definition for the term agency is "the state of being in action or exerting power". These definitions indicate that an agent can be physical or abstract in nature, but it must be able to act or cause effects as granted agency by, and, for others or itself [Sun99]. For example, in the financial market, security price is an agent when it affects trading behaviour. In turn, trading behaviour acts as an agent when it affects security price. The agency of an entity cannot be specified with reference to its intrinsic features, since it is so widely open and undetermined [Sun99]. For instance, the exact location at which the price of a commodity is displayed on a board may influence how much that commodity is traded. It is very important to draw a clear distinction between the term agent used in the EM, and the term intelligent agent used in modern computer science. An intelligent agent is a computer system that is capable of flexible autonomous action in order to meet its design objectives. Flexibility means that the system must be responsive to change in the environment, proactive in goal directed

[1] A behaviour is a reliable repetitive pattern of action

behaviour, and able to interact with other artificial agents and humans [Jen97]. The difference between an agent in EM and an intelligent agent as defined by Jennings (1997), is in the degree of human intervention to play the agency role, and the openness of the agent action. The behaviour and functions of an autonomous intelligent agent are preconceived and well formulated in advance. An agent action in EM is situated[2] in nature and emerges from the modeller's private insight and perception of the real world.

- *Definitions to maintain state:* A definition in EM is similar in character to a formula in a spreadsheet (cf. Figure 3.1). Any change to the value of a dependee (a parameter of the built-in function used) will give rise to a re-evaluation of the dependent variable [Yun92].

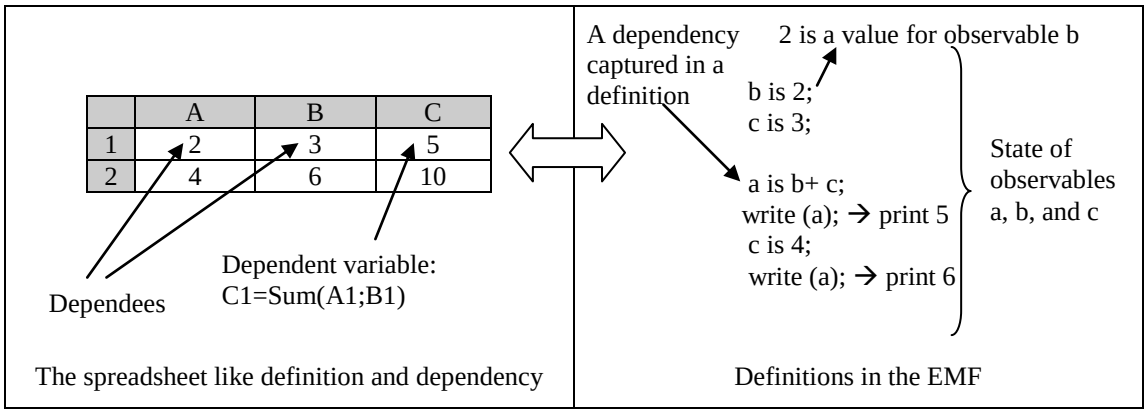


Figure 3.1 State and state representation in EM

State and definitive representation of state is illustrated by a simple two variables ordinary least square regression model developed using tkeden EM tool (refer to chapter 3 in the thesis web page on the Thesis CD). Definitions are used to maintain the state of the OLS estimates and residual errors represented by the observables beta_estimates, alpha_estimates, standard_error_alpha, and standard_error_beta. Part of the definitive script representing state in the OLS regression model is shown in the table below.

[2] An action is situated if it involves a conscious reference to context and choice of course of action. An action is not regarded as situated if it takes the form of a prescribed response or if it is an unconscious automatic response [Suc87].

```
%eden
Y is[10,15,36,18,19,20];  /* dependent variable taken as the annual
excess return on stock A*/
X is [12,56,78,98,90,67];  /* explanatory variable taken as the
annual excess market return  Rm*/

func sum_list_elements {
   para l;
   auto i, result;
   result=0;
   for (i=1;i<=l#;i++)
     result = result + l[i];
   return result;
}

func sum_square_list_elements {
   para l;
   auto i, result;
   result=0;
   for (i=1;i<=l#;i++)
     result = result + l[i]*l[i];
   return result;
}

func product_twolists_elements {
   para l,k;
   auto i, result;
   result=0;
   for (i=1;i<=l#;i++)
     result = result + l[i]*k[i];
   return result;
}


sumX is sum_list_elements(X);
sumY is sum_list_elements(Y);
sumsqrX is sum_square_list_elements(X);
sumsqrY is sum_square_list_elements(Y);
productXY is product_twolists_elements(X,Y);


/* computation of basic building blocks */
sumsqrx is sumsqrX -(1.0/(X#))*sumX*sumX;
sumsqry is sumsqrY -(1.0/(X#))*sumY*sumY;
sumxy is productXY -(1.0/(X#))*sumX*sumY;


/* computation of OLS estimates */
beta_estimate is sumxy/sumsqrx;
alpha_estimate is sumY/X# - beta_estimate*sumX/X#;

/* computation of the residual error */
```

Some basic functions operating on a list of elements, and returning a resulting list or number

A definition of observable as a value returned by a function

A definition of an observable as a formula

```
sum_square_residual is sumsqry - beta_estimate*sumxy;
s_square is sum_square_residual/(X#-2);

/* computation of the error in the estimate for the regression
parameters */
standard_error_alpha is s_square*sumsqrX/(X#*sumsqrx);
standard_error_beta is s_square/sumsqrx;
```

Table 3.2 Eden script for OLS Regression

The above model illustrates the use of a definitive script in EM and the similarity between dependencies in an Eden script and dependencies in a spreadsheet model.

Interaction with the script is open-ended. The dependent and independent variables can be changed and the list size can be increased by accepting the following re-definitions in the tkeden interpreter (cf. Figure 3.2).

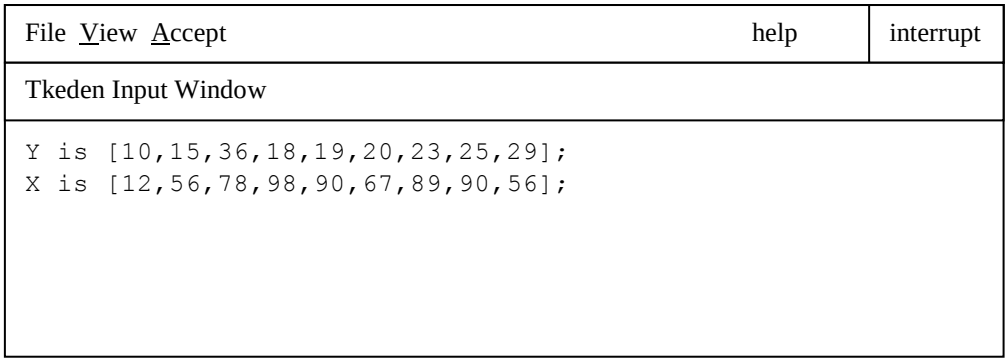


Figure 3.2 Introducing a definition / re-definition

Some advantages are gained when implementing OLS regression as a definitive script. First is the flexibility to change the values and size of the dependent and explanatory variables Y and X without the need to invoke any process to re-evaluate the estimation of the intercept and slope of the regression line. In an Excel spreadsheet, extending the range of the dependent and explanatory variables necessitate the use of the data analysis regression tool again with the new range of data. Compared to the use of a high level language like C, the Eden script is more flexible in the sense that a new definition of the dependent and explanatory variables Y and X can be accepted without having to include again the whole definitive script. The dependencies in the previously accepted script are maintained as long as they are not broken. This gives an added value over an implementation using a high level language, which requires running the program again with a new data set for X and Y.

The Eden script can be described as a radical generalisation of the spreadsheet concept in three respects. The first is in presentation because the dependencies are not only between

variables in tabular format (cf. values in spreadsheet cells) but can be across any observable within the system. The second concerns the underlying data type because we can use abstract data types representing a far wider range than in a spreadsheet. The third is agency which allows dependencies to be handled by many human or automated agents concurrently [Roe00]. Establishing a link between the Eden script of the OLS regression and the graphical representation of the regression line would demonstrate the generalisation of the spreadsheet concept beyond data in tabular format.

Depicting the efficient frontier[3] composed of portfolios with optimum risk-return trade-off using EM tools gives insight into the generalization of the spreadsheet dependency concept to apply to visual elements. A fuller description of the model related to portfolio diversification theory and the efficient frontier is found in the home page of chapter 3 in the thesis web page on the Thesis CD.

The definitive script establishes dependencies between observables that typically have some form of visualisation (point, text message, window, 2D visual metaphor) attached to them. The definitive script shapes the semantics of the visual elements attached to it.

The Eden notation is used to develop a script establishing dependencies between observables. The Donald notation is used for graph drawing, and the Scout notation is used for screen display. The script applies for a portfolio of two assets. Extension of this script to account for more assets would need additional definitions for the calculation of variance and covariance of returns.

The two classes of assets (their expected returns and probability of occurrence of these returns) are represented in the abstract data type list structure in the Eden notation:

```
/* r=[[return , probability of its occurrence in state 1 of the economy],..];*/
R1=[[20,0.2],[5,0.6],[-10,0.2]];
R2=[[50,0.2],[15,0.6],[-20,0.2]];
Rf= 12; /*the risk free rate of return*/
```

[3] This relates to portfolio theory first developed by Harry Markowitz in 1952. The thinking behind the explanation of the risk-reducing effect of spreading investment across a range of assets is that in a portfolio unexpected bad news concerning one company will be compensated for to some extent by unexpected good news about another. Markowitz provided us with the tools for identifying portfolios that give the highest return for a particular level of risk. Investors can select the optimum risk-return trade-off for themselves depending on the extent of personal risk aversion [BKM96, Arn98, EG95, CLM97].

Functions for the mean[4], variance[5], covariance[6] and correlation coefficient of returns are developed. The variance matrix and its inverse are formed. The efficient frontier[7] is a scatter graph relating portfolio returns (y axis) and portfolio variances (x-axis).

The Eden model developed allows the exploration of different return/variance combinations. Additional script can be easily added to visualise the security market line[8], and capital market line[9]. All observables are linked to each other by dependency relationships. New values given to observables will automatically update all the dependent observables as well as the graph.

```
invsigma11 is sigma22/(sigma11*sigma22 - sigma12*sigma21);
invsigma12 is sigma12/(sigma12*sigma21 - sigma11*sigma22);
invsigma21 is sigma21/(sigma12*sigma21 - sigma11*sigma22);
invsigma22 is sigma11/(sigma11*sigma22 - sigma12*sigma21);
inverse_sigma_matrix is
[[invsigma11,invsigma12],[invsigma21,invsigma22]];
```

Any observable value, can be queried at any time. A visualisation is attached to the two observables portfolio returns and portfolio variances that are linked by the dependency relationship defined by the efficient frontier equation. The definitive script establishing a dependency between Eden observable and Donald variables that define the shape and the semantic of visual elements is presented in the table below.

---

[4] $\bar{r} = \sum_{i=1}^{n} r_i p_i$ where $\bar{r}$ is the mean return; r_i is the return of the security if event i occurs; and p_i the probability of occurrence of event i.

[5] $\sigma = \sqrt{\sum_{i=1}^{n} (r_i - \bar{r})^2 p_i}$ where σ is the variance of the security ; $\bar{r}$ is the mean return of the security; r_i is the return of the security if event i occurs; and p_i the probability of occurrence of event i.

[6] $\mathrm{cov}(R_A, R_B) = \sum_{i=1}^{n} (RA_i - \overline{RA})(RB_i - \overline{RB}) * p_i$

where $\mathrm{cov}(R_A, R_B)$ is the covariance of the two securities A and B ; RA_i and RB_i are the returns of securities A and B if condition i occurs; $\overline{RA}$ and $\overline{RB}$ are the mean returns of securities A and B; p_i is the probability of occurrence of condition i

[7] $\sigma_p^2 = \frac{(\mu_p - r_f)^2}{Ar_f^2 - 2Br_f + C}$ where σ_p is portfolio variance; r_f is the risk free rate of return; μ_p is the portfolio return; A, B, and C are calculated from a unique combination of the risk-free asset and the "tangency portfolio" which maximises the expected utility of the investor's end-of-period wealth.

[8] A linear line showing the relationship between systematic risk and expected rates of return for individual assets (securities). According to the capital asset pricing model, the return above the risk free rate of return or a risky asset is equal to the risk premium for the market portfolio multiplied by the beta coefficient.

```
%eden
B is invsig1[1]*mu1+invsig1[2]*mu2;
A is invsig1[1]+invsig1[2];
C is mu1*invsigr[1]+mu2*invsigr[2];;
D is A*C - B*B;
func efficient
{
para mu;
auto va;
val=sqrt(((A*mu*mu)-2*B*mu+C)/D);
return val;}
proc constructefficient
{auto val;
MUy=[0,0.05,0.1,0.15,0.2,0.25];
Sigmax=[];
for (i=1;i<=MUy#;i++)
{
  val=efficient(MUy[i]);
  Sigmax=Sigmax//[val];
  }}
constructefficient();
  P is Sigmax;
  Std is Muy;

%donald
viewport drawgraph
int n,xsc,xsh,ysc,ysh
n=5
xsc=100
ysc=50
xsh=50
ysh=50
graph main, xaxis, yaxis
within main {
  x<I>=getx!( <i>)

  f<I> = gety!(<i>)
  nSegment = ~/n
  node = [circle:circle({x<i> *~/xsc+~/xsh, f<i> *~/ysc+~/ysh}, 10)]
  segment = [line:[{x<i-1>*~/xsc+~/xsh, f<i-
1>*~/ysc+~/ysh},{x<i>*~/xsc+ ~/xsh, f<i>*~/ysc+~/ysh}]]
           }

%eden
proc graphChange : std,P

{maxx=findmaxx();
 minx=findminx();
 maxy=findmaxy();
 miny=findminy();
 _n is (P# );
 scalexy();
```

[9] The set of risk-return combinations available by combining the market portfolio with risk free borrowing or lending.

```
 }

 graphChange();
```

Table 3.3 Eden, Donald, and Scout scripts for the CAPM

The following figure shows the screen display of the efficient frontier linked to observables in the model.

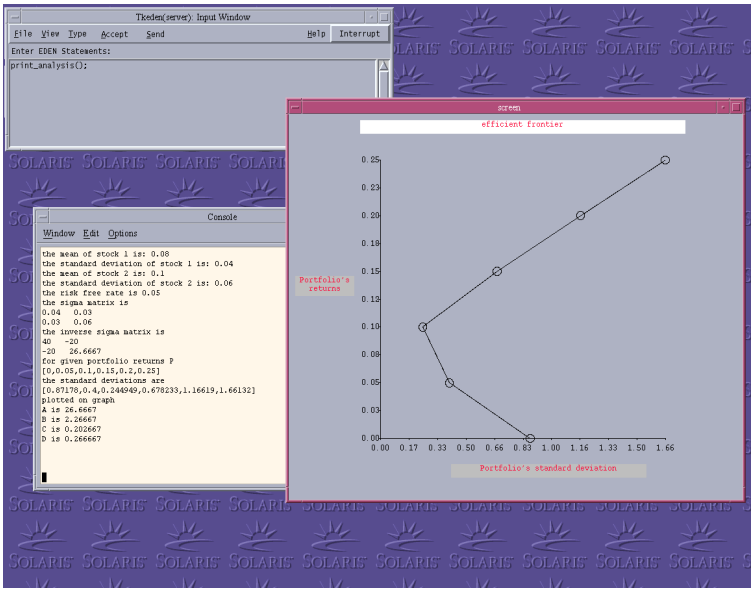


Figure 3.3 Use of Eden, Donald and Scout to explore the efficient frontier

- *Agent oriented analysis:* Agent oriented[10] analysis in EM relates the interaction between agents - in the first instance - to basic perception of observables. More sophisticated issues of knowledge representation can follow from this preliminary agent oriented analysis after the identification of a reliable and persistent mode of interaction between agents. For example, in a financial market context, a trader would resort to an agent-oriented analysis to gain basic knowledge of the behaviour and interaction in the market. Once this basic knowledge is established, more sophisticated mathematical modelling would be more appropriate. This perspective on financial modelling is consistent with Gooding's (1990) point of view on the importance of observation and experimentation in testing the truth and validity of new theories and technologies. The concept of agent

[10] Note that the agent oriented approach in the EMF differs from agent orientation as referenced in AI research such as [Sho90].

action in EM is complementary to the notion of indivisible change of system state as expressed in definitive scripts [BJ94]. State in EM is represented by means of a system of definitions, each of which defines the value of a variable either explicitly, or implicitly in terms of other variables and constants. Transitions from state to state are performed by redefining variables [BNRSYY89].

3.1.2 EM techniques

The concepts of observables, agency, dependency, definitive representation of state and agent oriented analysis in EM support experiential knowledge construction of an application domain by developing a computer artefact / a cognitive artefact based on construing a situation and constructing an associated Interactive Situation Model (ISM).

A literal dictionary definition [rhyme] of the term construal is "an interpretation of the meaning of something". In EM, a construal is represented metaphorically via a physical artefact, typically computer-based, and has a number of key features [Bey99]:

- *It is empirically established (it is informed by past experience and is subject to modification in the light of future experience);*
- *It is experimentally mediated;*
- *The choice of agents is pragmatic (what is deemed to be an agent may be shaped by the context for our investigation of the system); It only accounts for changes of state in the system to a limited degree (the future states of the system are not circumscribed).*

This interpretation of construal has a similar meaning to that introduced by Gooding (1990):

> *"Construals are a means of interpreting unfamiliar experience and communicating one's trial interpretations. Construals are practical, situational and often concrete. They belong to the pre-verbal context of ostensive practices".*

Beynon (1999) identifies an important difference between construing a system in the EM framework and in the AI framework. In the AI framework, a system is construed as "acting as if it were inferring", a mathematical structure of objects is adopted, and preconceived functions for a system to achieve its purposes are presumed. A construal in EM is represented metaphorically via a computer-based physical artefact constructed based on previous experience and is subject to exceptional behaviour for which there is no pre-conceived explanation. In the Empirical Modelling framework a system is construed, "as if it were

composed of a family of agents, responding to observables, and exercising privileges to change their values in the context of a set of dependencies between observables”

A literal dictionary definition [rhyme] of the term situation is “a condition or position in which you find yourself”. In EM, activities are considered as “situated”. The term "situated activity" introduced in [Sun99] refers to a coherent sequence of situated actions that is constructed by the interaction between a human agent and its environment[11]. In our daily life, we envisage many situated activities. Examples of these are illustrated by considering different scenarios drawn from a financial context:

- buying a portfolio of shares
- visualising a financial data set
- joining a group conversation on a debatable issue in finance such as market efficiency.

In the first scenario, the investor might conduct a financial market analysis to support his/her stock selection, consult a broker and delegate to him/her the task of portfolio construction, or make a random selection of shares. In the second scenario, the investor might be satisfied with a rough paper sketch, resort to graph drawing utilities in a spreadsheet application, use a special purpose graphic package, or write code in a high level programming language to visualise the data set. The third scenario involves listening and replying to the speaker.

An action is situated if it involves a conscious reference to a context and a choice of a course of action. A situated activity is different in character from an activity specified by a formal algorithm. A situated activity is difficult to prescribe in advance. This difficulty is revealed by attempting to answer, in advance and with full certainty, the question of how this will be done in each of the above considered scenarios.

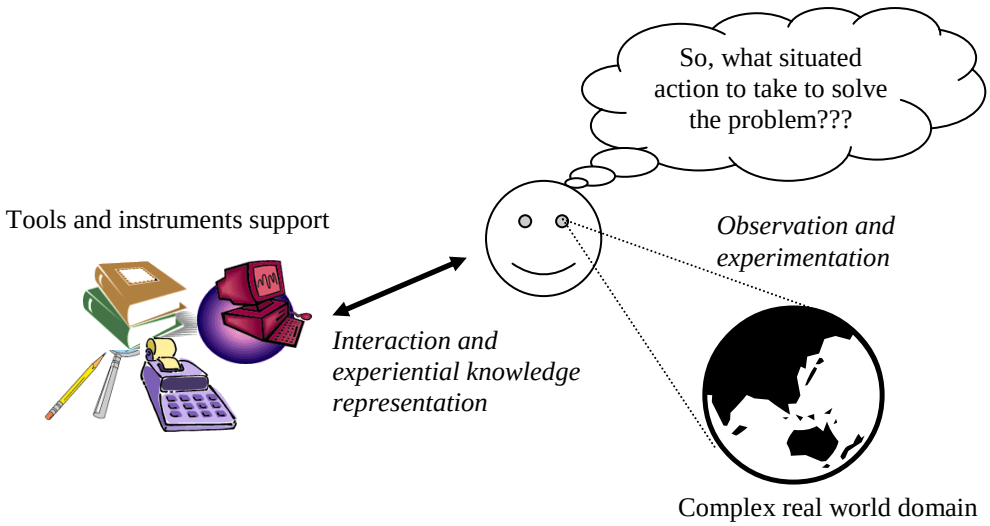


Figure 3.4 Solving problems in the real world domain

Solving problems in the real world domain is a situated activity rather than a formal activity, especially when problems are first encountered. Suchman (1987) argues that most plans (algorithms, strict laws, formal methods) used by human agents serve as a resource rather than a source of control in everyday life (cf. Figure 3.4). This argument is also supported by the fact that a solution to a real world problem is context dependent (i.e. it cannot be detached from the real world and abstracted in a formal algorithm), and human centred (i.e. the human role is central, is not pre-conceived at an initial stage, and is difficult to capture in formal logic or rules).

However, a situated activity is error prone because it relies upon human discretion, as compared to a formalised process derived from an engineering discipline. Hale (1998), Norman (1983), and Radford et al (1974), recognise the fact that the human being is inevitably error prone and forgetful, learns slowly from experience, and can be seriously distracted by the external environment. These human factors highly influence the structure of the situated activity. The development of tools and instruments aims at supporting the human activity in reducing the impact of human weaknesses. However, in the case of computer-based tools, formalized interaction and context-independent algorithms limit the effectiveness of these tools in empowering human strengths.

The EM technique of construing a situation entails construing a system in a situated way. That is a system admits different construals, each formed based on a situated judgement. An Interactive Situation Model (ISM) is developed to explore different construals. The ISM is a computer-based environment constructed through a situated modelling activity. Unlike a closed-world computer model with a fixed interface, an ISM is always open to elaboration and unconstrained exploratory interaction [Bey94]. States within the ISM metaphorically represent pertinent situations from the application domain, and possible transitions between states are explicitly constructed so as to be consistent with the developer's construal of a system in terms of agents, observables, and dependencies [BCSW99].

Interactive Situation Models were first introduced and used in EM to assist in the software development process, and in particular in the requirement engineering phase of this process. The use of an ISM was further extended to support the development of reactive systems, and for providing computer-based support for diverse activities in the real world domain

[11] Refers to the external surroundings of an individual.

including: business process modelling, product design, learning, decision support, and interpersonal communication.

The use of an ISM is illustrated in this chapter, and in chapter 5, with reference to the story of a retail trade in NYSE extracted from [Har98]. The purpose of the ISM developed for this story is the exploration of the trading mechanisms in the financial market and the design and organization of financial markets. The trading story can be referred to in the home page of chapter 3 in the thesis web page on the Thesis CD.

Constructing an ISM for a NYSE retail trade is a way of modelling an external observer's explanation of the retail trade process (RTP). In its most naïve form, such an explanation explicitly relates the actions of agents to the stages of the trading protocol. This simply involves identifying the actions for which each agent is responsible, and identifying the preconditions under which each action is performed. The major roles in a retail trade are played by the investor and the broker. The *investor* requests information on a particular stock from the broker, puts a trading order, confirms his order, pays for his transaction, and acquires or releases share ownership following the execution of his order. The *broker* requests quotes from the quote information system, returns this information to the investor, enters any received order in the order entry system, reviews the order details prior to its release in the order entry system, reports the trade execution to the investor, receives payment including charge fees, and mediates the exchange of share ownership. Each of these actions on the part of investor and broker is performed at a specific stage in the RTP. The following figure is a screen snapshot of an ISM built using the EDEN interpreter.

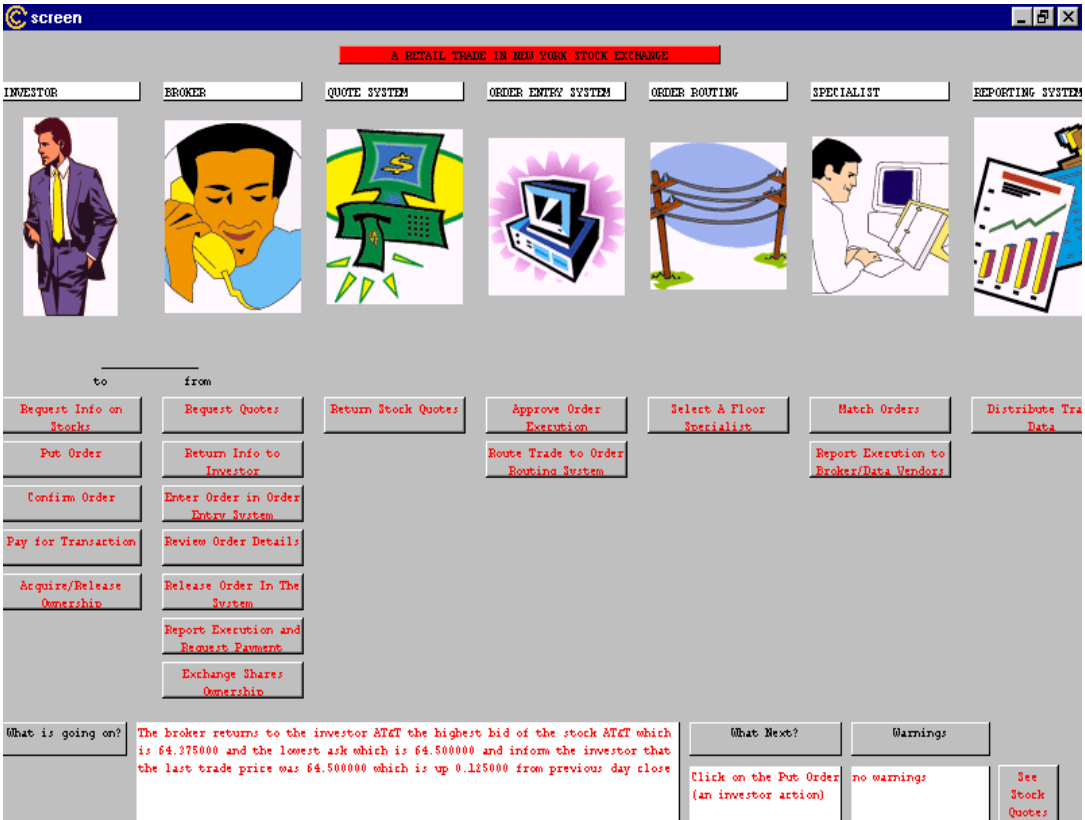


Figure 3.5 A snapshot of the ISM for a retail trade in NYSE

3.1.3 EM notations

In Empirical Modelling, all state-changing activity is attributed to agents. An agent can be a human actor, a state-changing process or component. The role of an agent can be played by a modeller, or by the computer. In the process of explanatory analysis of a situation, many aspects of the agency and dependency that are being identified will be implicit in the interaction between the modellers and the computer artefact. In understanding the behaviour of a system, it is also typically important to identify explicit protocols and stimulus-response patterns that are characteristic of agent interaction [Bey99, BM00].

A special-purpose notation - the LSD notation - has been introduced to describe such agency. An *LSD account* is a classification of observables from the perspective of an observer, detailing where appropriate:

- the observables whose values can act as stimuli for an agent (its *oracles*);
- those which can be redefined by the agent in its responses (its *handles*);
- those observables whose existence is intrinsically associated with the agent (its *states*);
- those indivisible relationships between observables that are characteristic of the interface between the agent and its environment (its *derivates*).
- what privileges an agent has for state-changing action (its *protocol*).

The use of an LSD account is illustrated with reference to the model of a retail trade in New York Stock Exchange considered earlier.

The roles of the various agents in the NYSE have to be understood in terms of the relevant observables. Some of these observables (such as the current status of a BUY/SELL order) are particular to the retail trade situation, but the actions of agents also relate to observables generic to the online trading context.

In the online trading context, the social network comprises investors, brokers, dealers, arbitrageurs, and boards of trade. The trading marketplace may be a physical trading floor or an electronic system. In the retail trade situation, the relevant ***agents*** in the model are identified as: the *investor*, the *broker*, *the physical stock exchange*, *the company stock the*

quote information system, the *order entry system*, the *order routing system*, the *floor specialist*, and the *information reporting system*. The relevant ***observables*** for the participating agents comprise:

- *Order information*, including: investor name, ID, BUY/SELL order, share name and symbol, quantity of shares, type of order (such as market, stop loss, limit order, etc.), price (if needed), expiry date of the order, the date and time of the order.
- *Stock quotes*, including: stock symbol, bidder, BID/ASK, price, size, time and date.
- *Stock information*, including: stock symbol, stock name, last trade price, change from previous day close, time last traded, place last traded, highest day price, lowest day price, day volume.
- *Order indications from dealers and brokers*, including: the stock name, the name of the broker/dealer, the time, and the date.

A possible account of the broker's response to an information request might be:

1. check status of request information action by investor;
2. get investor information request;
3. direct information request to quote information system;

update current RTP status;

The current stage reached in the RTP is interpreted as an observable for the participating agents. Each agent action is formulated in terms of re-definitions of observables. For instance, in the initial stages of the RTP, the broker requests quotes from the quote information system when an investor has requested information on a particular stock.

The following table presents the LSD notation used to describe the agency and dependency in the account of the story of a retail trade in New York Stock Exchange.

The LSD template for describing the broker agency in the account of a retail trade in New York Stock Exchange	
agent broker {	
state	info_requested, quotes_info_requested, Commission rate, bid and ask price (if the broker acts as a market maker or dealer), trade history, personal account (profit account = cumulated commission revenue + revenue from spread (in case broker is dealer)
oracle	stage_in_retail_trade info_requested Investor's orders, price change, order status
handle	order status, bid and ask prices of a stock, portfolio holding of investor, commission rate
derivate	stage_in_retail_trade = F(info_requested, …) ….

```
protocol   (stage_in_retail_trade = init_trade)
           and (info_requested) → quotes_info_requested=1
           (order_issued=1) → validate_order( )
           (order_approved=1) → order_directedtorouting system=1
           (order_directedtoroutingsystem=1) → order_directtofloorspecialist=1
}
```

Table 3.4 LSD account for the broker agent in the story of a retail trade in NYSE

3.1.4 EM Tools

The principal EM software tool developed so far is tkeden. Tkeden can interpret three types of notation: Eden, Scout, and Donald.

The modeller viewpoint is represented by a script of definitions (a *definitive script – Eden script*) resembling the system of definitions used to connect the cells of a spreadsheet. The variables on the LHS of such definitions are intended to represent observables associated with the external situation. There is typically some form of visualisation attached to them, so that for example a variable can denote a point, a text message or a window displayed on the computer screen. Scout and Donald notations are used to attach a 2D visualization to Eden script.

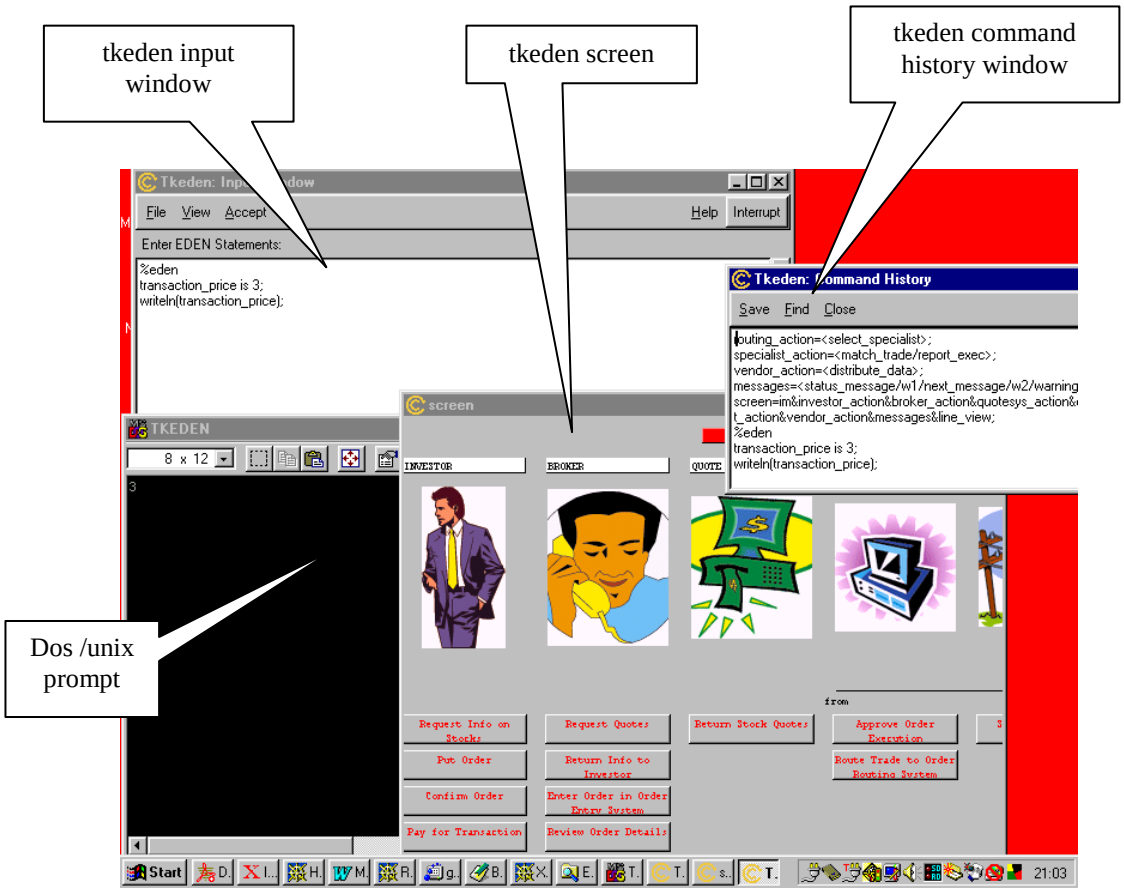


Figure 3.6 The tkeden interpreter

The tkeden interpreter is composed of the following components:

- An input window, where Eden, Scout, and Donald notations are edited and accepted.
- A command history window, that stores all commands accepted by the tkeden input window since the start of the current modelling session.
- A screen, where Scout and Donald notations are visualized using 2D geometric metaphors including buttons, text input, lines, and circles.
- An output window for displaying the definitions and values of variables.

A distributed version of tkeden has also been developed: dtkeden. It is implemented on a client-server architecture, in which the viewpoints of individual modellers are represented by independent definitive scripts executed on different client workstations. State changes are communicated between clients by sending redefinitions across the network via the server. Communication strategies can be specified via the server to suit different purposes. The server can play a role in negotiation between clients, resolving conflicts or dictating the pattern of interaction and privileges of modellers.

Distributed Empirical Modelling is illustrated with reference to the case study of a distributed Stock Market Game. The game was originally developed by the author, and tailored to the style of trading in London Stock Exchange by Ajul Shah. This development of the game is different in spirit from a game theoretic approach[12]. Whereas game theory is more sophisticated in analysing multi-person decision making as described in [Gib92], it gives limited scope for experiential knowledge and personal insight.

Observation and experiential knowledge are used to develop the game. The identification of players in the game is subjective and reflects personal insight and basic understanding of trading behaviour. The development of this game illustrates a basic application of Distributed Empirical Modelling (DEM) technology to modelling financial markets. A very simple model

[12] Game theory is the study of multi-person decision problems. Such problems arise frequently in economics. At the micro level, models of trading processes (such as bargaining and auction moodels) involve game theory. Labor and financial economics include game-theoretic models of the behaviour of a firm in its input market. There are also multi person problem in a firm. Games are classified into classes: static games of complete information, dynamic games of complete information, static games of incomplete information and dynamic games of incomplete information. Four notions of equilibrium are in these games: Nash equilibrium, subgame-perfect Nash equilibrium, Bayesian Nash equilibrium, and perfect Bayesian equilibrium [Gib92].

of electronic trading emerged with little support for the understanding of some basic aspects of stock trading. The initial model does not take account of trading rules and regulations in any market. A distributed version of tkeden, dtkeden, is used to implement the model on four workstations.

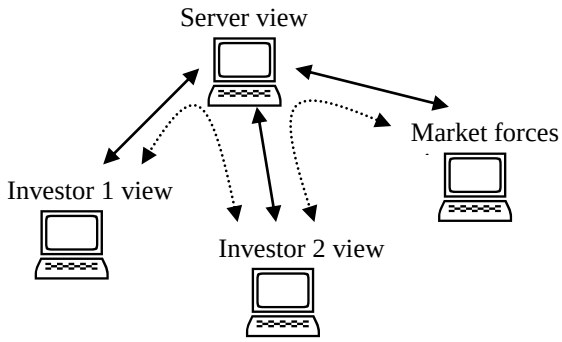


Figure 3.7 The star configuration of the distributed game

A star-type configuration for network communication (cf. Figure 3.7) is adopted to link a server to three clients representing two investors and the action of market forces. Each investor can monitor the prices of the securities within their view. They can trade (buy/sell) these securities. Some business rules in trading are to be respected (an investor cannot sell a share that he does not own; an investor cannot buy a number of shares of a particular firm exceeding the number of shares issued by the firm). The **`market forces`** agent simulates all the events affecting the prices of traded shares and the decision made by listed firms to issue an additional number of shares. The **`market forces`** agent can change the price of the shares up or down and it can increase the number of issued shares by a given firm. Each investor is supposed to construct his/her own portfolio by buying and selling shares from the market. An intelligent investor would buy shares if shares are under-priced and would sell shares if they are over-priced. The portfolio balance of an investor is the market value of all the shares within his/her ownership. The financial position of investors is the sum of their cash holding and the market value of their portfolio, which is subject to change under the action of the **`market forces`** agent. The **`market forces`** agent can increase the number of issued shares by a firm thus giving the investor an opportunity to buy a larger number of shares issued by a particular firm. A sample of the client-server communication is shown in Figure 3.8 below. It illustrates distributed communication through definitive scripts.

The **`market forces`** agent sends a new value for the price of share 1 to the client **`investorI1`**. This will change the value of **investorI1_balance** that depends on the value of the price of share 1 via a definition.

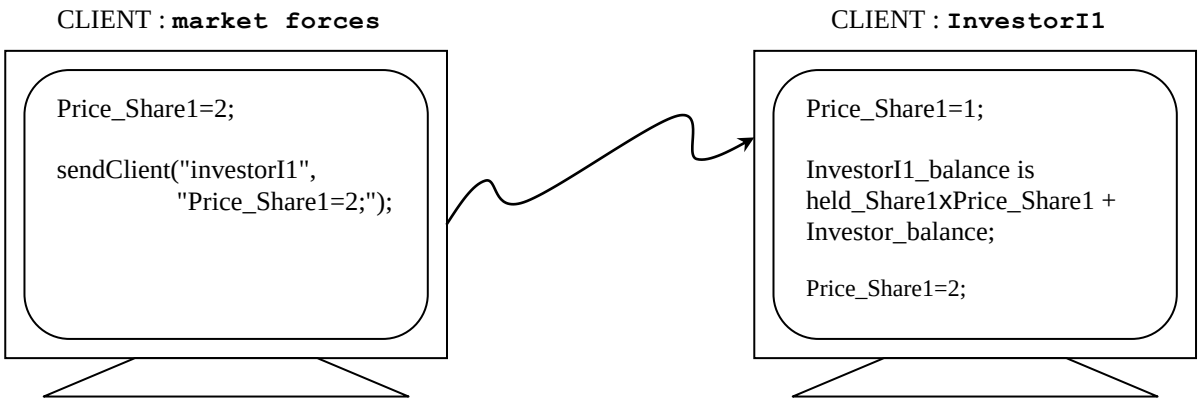


Figure 3.8 Distributed communication of definitive scripts

Where interaction is concerned, dtkeden allows transmission of re-definitions between scripts located at the server or the clients according to the star-type configuration for network communication. The semantics of such a re-definition is wide open: it encompasses actions that assign different values to observables or create dependencies between these observables. New observables can be introduced, such as might represent external factors affecting market behaviour. The Scout interface handles text-based interactions, for example, for buying and selling shares and for displaying the winner/loser. The most primitive but powerful mode of interaction with an EDEN interpreter is through entering definitions directly into an input window. In principle, all market participants can exploit input of this nature. The investors can send messages to each other via the server, and the server can broadcast general news to all market participants. It is possible to specify which observables values are **handles** and **oracles** in LSD terms, and accordingly then can be changed and/or inspected by a market participant [BM00, Sun99].

The quality of communication in dtkeden stems from the fact that the representation of state is definitive, transitions are effected by transmitting re-definitions and each interpreter actively maintains and monitors the current state of relevant observables. All these features reflect the way in which agents are construed to act and interact in the real-world. They are

also relevant to architectures for agency. Where present computing platforms are concerned, the need to distribute EDEN interpreters may be regarded as imposing a significant computational demand on each client. There are ways in which this issue could be addressed – in particular, by localising and customising dependency maintenance to balance the resources available for reconstruction of state and the bandwidth for transmission of state information.

A comparison with other programming technologies that might be applied to model virtual trading is helpful. Current web technology is highly document-centric. To transmit non-textual data requires techniques such as pre-process-and-publish that are far from delivering the direct influence over remote state that the transmission of a re-definition effects. The scope for interactive agency in a web network is inhibited by the standard net protocols: the state of a webpage is updated only when the viewer of the webpage initiates a request. With conventional software development methods, a Java implementation of virtual trading would be targeted at a specific preconceived requirement for market participant interaction. In this connection, there is a trade-off between the narrowness of the requirement and the quality and efficiency of the solution.

A brief account of how the model is constructed illustrates how its functionality remains open-ended. As in all EM models, there are many different ways in which the constituent definitive scripts, functions and actions can be organised into clusters. These can correspond to conceptual layers in the model, to submodels suitable for re-use, or to partitions into observables associated with specific sub-objects for instance.

The above simple distributed model simulates the behaviour of an uninformed[13] investor in an inefficient market. The market is inefficient because current prices do not necessarily reflect publicly available information. The investor is prone to high loss as well as high gain on a speculative basis. However, the price reporting and trade execution process is highly efficient, as current prices are directly appearing on the screens of the investor, and the transaction is executed automatically without delay. This relates to research in finance in the area of market microstructures. The use of EM technology to model financial market microstructure will be considered more deeply in chapter 6.

The model so far developed is unrealistically simple. In the real world, the delay in transmitting actual price changes to investors, the delay in trade clearing and settlement, and the lack of market transparency, all affect the trade process. The model can be extended to

[13] An uninformed investor is an investor who does not perform any analysis before executing any trading transaction.

account for all these factors. Moreover, decision support for the investor and intelligent analysis can be incorporated in a more advanced model.

Shah (2000) introduced some improvement to the model, bringing it closer to real trading in a specific exchange and extending its capacity (more investors can connect to the server, more stocks are considered, a larger source of market information is provided, and additional visualisation is introduced for the investor's portfolio and financial indicators). The model is a prototype for a multi-player game in which each player has the ability to analyse information and then buy and sell shares. The mechanism of the trade is more realistic. Trading rules for order matching are followed. A model simulating SETS[14] is also developed separately.

3.2 Distinctive qualities of Model Building in EM

The principles, techniques, notations and tools of EM exhibit the following distinctive characteristics:

a) the focus on state as experienced;

b) the maintenance of a semantic relationship between an application domain and a computer-based artefact;

c) the use of an artefact for knowledge construction;

d) the use of definitive scripts to support collaborative distributed modeling.

a) Empirical Modelling focuses on state as experienced rather than state as abstracted

State in EM vs state in mathematical/ conventional model

There are key differences between the representation of state in EM and in abstract mathematical / conventional models. These differences are attributed to several factors and considerations related to: i. entities in the model; ii. relationships between entities in the model; iii. human agency; and iv. the scope for distinguishing different aspects of state

[14] SETS (Stock Exchange Electronic Trading Service) is the LSE's fully electronic order book trading mechanism. The order book is the central price formation and trading mechanism for the securities in the FTSE-100 index, reserve securities and others. There are no market maker quotes for these securities. The order book allows participants to submit orders displaying their willingness to buy or sell share at specific prices, or to execute against displayed orders. Execution occurs when a buy and a sell order match. Orders are submitted by stockbrokers either for clients or for themselves.

i. The primitive entities in EM are observables that have counterparts in the real world. Entities in mathematical / conventional models are variables that are abstract representations of observables in the real world. Once identified, they constitute the basic elements of the abstract representation of a real world domain.

In EM, observables can be added to the computer-based model to reflect our growing ***experiential*** knowledge of new observables in the real world. There is no preconceived description of, nor limit on, the number of observables in an EM model. In mathematical / conventional models, abstraction involves focusing on particular aspects of the real world domain and on simplifying reality for ease of understanding and representation. This limits the number of variables in mathematical / conventional models.

Observables in EM can be agents (instigator of state change), oracles (seen by other agents) or handles (manipulated by other agents). No particular characteristics can be attributed to the variables in a mathematical model apart from having a particular value determined by the specific function that they are introduced to serve.

ii. In EM, relationships between entities (observables) are established through definitions to reflect particular relationships between observables in the real world. These do not take the form of absolute invariant relationships between values of variables but of dependencies that express the modeller's current provisional expectations about how changes to some observables indivisibly affect the value of other observables. In particular, relationships between observables in the real world are not permanent in time. This reflects two factors: first, our knowledge of the relationship between observables in the real world is not perfectly exact; and second, observables in the real world may undergo change, and hence their relationship might change as well. As such, relationships between observables in EM can be altered by new definitions or re-definitions to reflect change encountered in the real world. In a mathematical / conventional model, relationships between variables are pre-conceived and formally established. There is no point in adding new relationships or altering existing ones as the determination of these relationships is strictly defined by the abstract representation of the real world.

iii. Human agency is central in EM. An EM model assumes the existence of a super-agent (the modeller – a human) who observes and interacts with the real world and the computer-based model. This super-agent sees the real world from a personal subjective view and constructs the computer-based model according to his/her own

view. The growing experiential knowledge of the human modeller of the real world domain enriches the model with observables and relationships between observables and the identification of agents and agent actions. Observables, relationships between observables, and agents in the model are determined subjectively by the human modeller and are subject to change with growing knowledge and evolving modes of observation. In a mathematical / conventional model, variables and relationships between variables are objectively determined to reflect an abstract representation of the real world that is based upon preconceived modes of observation and interpretation. Human intervention is permissible to change the values of certain variables following prescribed actions and through appropriate interfaces.

iv. State in EM is subjective (reflecting the modeller's view of the external world), situated (reflecting the actual situation in the real world) and context dependent (strongly related to the real world context to which it refers). As such, state in EM reflects state as experienced in the real world. State in a mathematical conventional model is an abstract state detached from its real world context. It hardly permits experiential knowledge construction.

Illustration The interaction with an EM model is open ended and resembles our interaction with the real world. This is illustrated with reference to an example[15] of the determination of the price of a security in the financial market, where P refers to the price of the security, DD to its demand, and SS to its supply. In the context of this example, an EM model reflects how our understanding of the relationship between P, DD, and SS can evolve. The modeller might regard DD and SS as things that are observed and that can be used in establishing P. He / she might think of DD and SS as agents that affect P according to some protocols and might identify a relationship between P, DD, and SS. Such a relationship can be re-defined to reflect evolving knowledge of P as new observables, such as *economic conditions*, are considered in the model.

In comparing the EM model with a mathematical / conventional model for this example, the differences are clearly drawn:

- Entities in the EM model are the observables P, DD, SS that are determined subjectively by a human modeller who is monitoring the state of the financial market and interacting with the computer-based model. Whereas entities in the mathematical / conventional

[15] The considered example is hypothetical and does not derive from a particular theoretical foundation in finance.

model are the variables P, DD, SS that are objectively determined in an abstract representation of the real world (financial market).

- Relationships between entities in the EM model are empirically established by the modeller following his / her experiential knowledge of the state of the financial market. Such relationships may take the form of definitions such as: **`P is f (DD, SS)`**. Relationships between variables in the mathematical / conventional model are determined by the abstract representation of the real world and take the form of an expression that can be evaluated according to the values of its variables : **`P= f (DD, SS).`**
- Observables in the EM model are classified as agents, handles, and / or oracles. DD and SS may be considered as agents affecting the state of P, and P as a handle and oracle for DD and SS. An LSD description can be used to capture the description of agency and dependency in the model. The LSD description for DD may take the form of:

```
Agent DD
{
   State  Value_of_DD
   Oracle P
   Handle P
}
```

  Variables in the conventional / mathematical model can only take on the values determined by the relationships in the abstract mathematical model.
- The state of the EM model is situated, context dependent, and subjectively determined by the modeller. Growing experiential knowledge of the modeller alters the state of the model. The modeller might conceive a new observable (economic condition) affecting the state of the model and introduce a re-definition in the model such as:

  `P is f (DD, SS, economic condition)`

  State in the mathematical / conventional model is determined by its behaviour (a repetitive abstract state).

b) Empirical Modelling aims at maintaining a semantic relationship between a computer-based artefact and an application domain that reflects the modeller's construal

Motivation

Empirical Modelling addresses the problem of the separation between experiences of the real world and of the computer-based model. Beynon (et al, 2000) argues that such a separation may be less of a problem in scientific or engineering applications where theories and abstract entities can be successfully applied to a certain extent. But in social and business domains such a separation leads to major problems. This stems from the difficulty of applying

contextual information appropriate to different situations and the difficulty of end-users[16] to modify the models.

The semantic relationship in EM

The focus on state as experienced and the continuous human engagement in the modelling activity in EM implies a semantic relationship between the computer-based model and its corresponding real world referent. Such a semantic relationship can potentially support semi-automated activities where human input and agency is paramount.

In EM, interaction with the computer-based model can be directly compared with real world experience. This interactive experience can be mediated by metaphors[17] or by a virtual reality style of interface[18]. The main objective of the computer-based model is to cultivate the understanding of the modeller of his experience in the real world. Experiential knowledge about the state in the application domain can be represented by experiential knowledge about the state in a virtual environment. A semantic relationship would then be established between the two experiences [BRR00-1]. Experiential knowledge about state in the application domain is subjective and needs a medium to expose it publicly, however, experiential knowledge about state in a virtual computer-based environment can be to some extent recorded by information processing mechanisms. This emphasises the use of the computer as an instrument for experimentation.

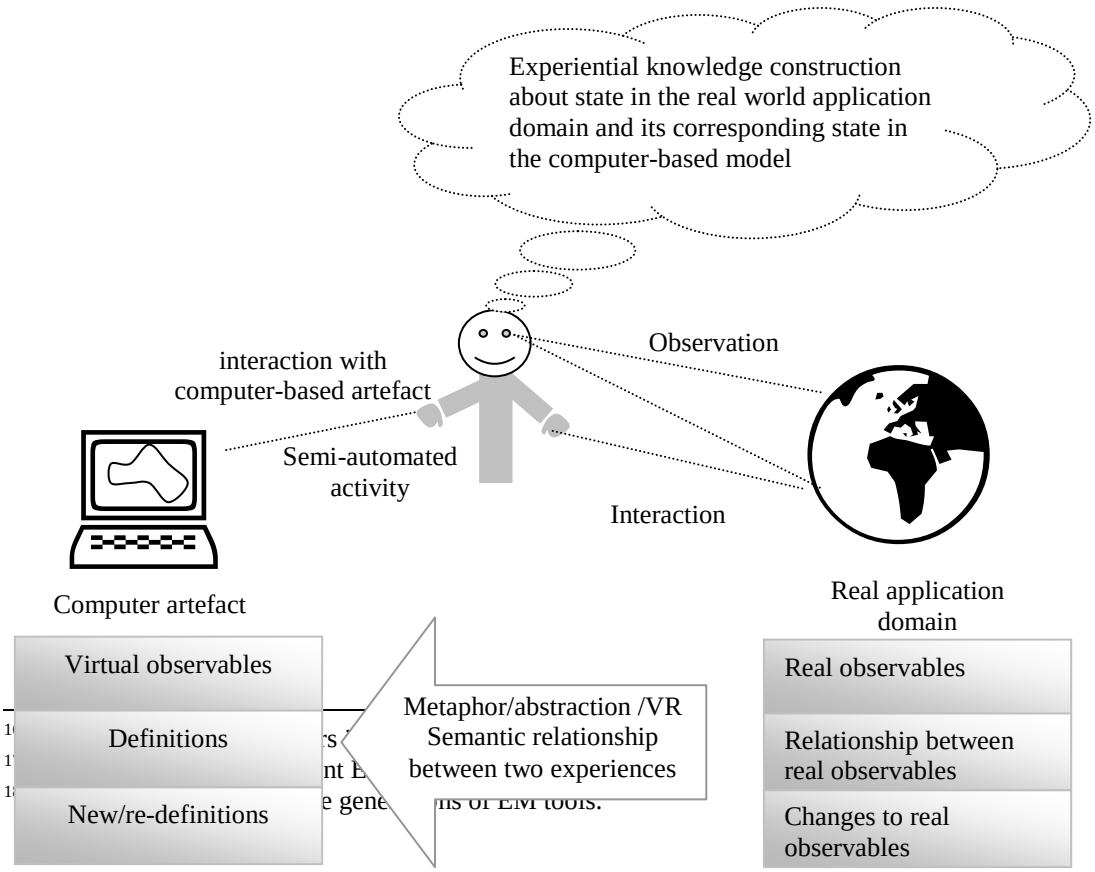


Figure 3.9 The semantic relationship between the computer-based artefact and its real world referent

Figure 3.9 illustrates the concept of representing experiential knowledge about state in the application domain by experiential knowledge about state in a virtual environment (EM computer-based artefact). A semantic relationship is established between observables in the real world domain and the computer-based artefact. Observables in the computer-based model are not abstract representations of entities in the real world but reflect the modeller's understanding and interpretation of real world observables. Relationships between observables in the real world are associated with definitions relating observables in the computer-based artefact. Changes to observables in the real world application domain correspond to new definitions or re-definitions relating observables in the computer-based model.

In EM, the construction of the artefact is informed by how the modeller understands the real-world situation surrounding the software system under development. Specifically, the artefact embodies the modeller's expectations concerning the agency that affects observables and dependencies between them ("the modeller's *construal*").

Technology support

Empirical Modelling technology suggests a broad foundation for computing that harnesses features of Virtual Reality, AI, and database technology to establish a virtual environment that is semantically related to its real world referent.

c) the use of artefact for experiential knowledge construction

Experience

Experience, literally defined in [rhyme] as "the accumulation of knowledge or skill that results from direct participation in events or activities", constitutes a ground base for truth and knowledge. Gooding (1990) asserts that experimentation is a hallmark of scientific activity, and attaches a great importance to the role of observation and human agency in practice.

Three issues are important in considering human experience in the real world:

- Acquiring experience
- Acknowledging an experience (being aware of the experience, and understanding it)

- Describing and sharing experience
- Learning from experience (personal subjective, public objective).

Knowledge

Knowledge is the psychological result of perception and learning and reasoning and can be implicit or explicit. While preceding knowledge, experience is broader, and ill structured. Knowledge is the outcome of experience that can be repeated with a certain degree of faithfulness. Reliable personal experience is subsequently translated into objective and structured knowledge that helps in identifying a reliable behaviour for a system solving problems in the application domain. Dienes (et al 1998) definition of knowledge indicates that knowledge is limited and concise, whereas experience is not limited in space and time. Experience is subjective, and human imagination, observation, and agency are paramount. Personal experiential knowledge is subjective, ill structured, incomplete, and continuously evolving.

Common practice for knowledge construction

In common practice, constructing experiential knowledge of a particular domain can be initiated by adopting one of the two approaches: (i) empirical testing of theories pertaining to the domain; (ii) following personal insights in observing and experimenting within the domain

Both approaches are viable and support experiential knowledge construction to some extent. The problem with the first approach is the validity of the underlying assumptions of the theories in all possible situations. The second approach might be considered a primitive one, and is likely to be adopted by the non-expert in the domain. Reconciling the two approaches is a challenging task, and requires a framework that encompasses the experimentation, the formulation, the testing, and the amendment of theories. The first challenge is that the theory is represented abstractly and admits no easy amendment. The second challenge is devising a computational framework that supports a broad activity that embraces theory construction, validation, and testing. The third challenge is the continuous change that makes theory formulation hardly possible.

Experiential knowledge construction in EM

EM aims at supporting experiential knowledge construction that can potentially lead to knowledge formulation following continuous interaction with the computer-based model and the discovery of a reliable repetitive pattern of interaction.

Traditional use of the computer has focused on representing experience in the real world through formal approaches. However, EM technology relies on the key concepts of observation, agency, dependency, agent oriented analysis, and definitive representation of state and state transition. It adopts the techniques of construing a situation, constructing

Interactive Situation Models (ISMs), metaphorically representing state through ISMs, and developing cognitive artefacts – typically computer-based. These key concepts and techniques, enacted using notations and computer tools, promise to deliver a framework (typically computer-based) that favours conducting experiments and gaining experiential knowledge of the real world. This framework aims at establishing a strong correspondence between real world activity and computer-based activity. Given the primary role of experience in knowledge construction and formulation, providing computer-based support to this real world activity serves to enrich an initial, yet essential, phase in computer activity referred to as requirements engineering. Research work on EM technology to date has served as proof of concept for supporting an ongoing requirements engineering activity that gradually and continuously feeds all other derived computer-based activities.

d) Empirical Modelling technology enables the communication of definitive scripts to support collaborative distributed modelling

Motivation

Modelling the real world as seen in the eyes of an external all-powerful human modeller has many drawbacks. These stem from several factors: i. the individual bias in the modeller's understanding and interpretation of phenomena in the real world; ii. the load on the modeller who is supposed to play the role of all agencies affecting the state of the model; iii. the lack of realism in the modelling activity; and iv. the foregone benefit of group social activity in modelling.

Collaborative modelling in EM

The Distributed Empirical Modelling framework aims at overcoming personal modelling by supporting collaborative[19] distributed modelling. This can potentially serve the objectives of:

- redressing the individual bias in the modelling activity
- restoring the balance in the modelling activity by inviting every agent (human and / or automatic) to take his/her/its role in the modelling activity through appropriate views and privileges for actions
- bringing more realism to the modelling activity by involving every participant in the real world domain (user / developer / designer in the context of software system development or manager / personnel in the context of the business or financial enterprise)

[19] Sun (et al, 1999) compares collaborative distributed activity to co-ordinative and subordinative. Collaborative distributed activity is situated and favour sharing insight in an open ended way.

- benefit from sharing insight and understanding in group social computer-based modelling

Empirical Modelling technology promises to support a situated group social activity by accommodating non-preconceived modes of interaction and taking account of the context and situation of the group social activity [Sun99].

The Distributed Empirical Modelling Framework

The distributed Empirical Modelling framework (DEM) as introduced in [Sun99] supports a modelling activity where "the system modelled can be observed from the perspective of its component agents and an objective viewpoint or mode of observation to account for the corporate effect of agent interaction is identified". The DEM supports collaborative relationships between modellers concerned with understanding that is socially distributed [SB99]. Such relationships engage with issues of subjectivity and objectivity associated with distributed cognition [Hut95] and common knowledge [Cro94, EM87]. In a collaborative relationship, there is no possibility of relying entirely upon closed-world representation and preconceived patterns of interaction [SB98]. Such interaction is situated intelligently and can only be planned in advance to a limited degree.

The aims of DEM as introduced in [BS99] are to examine the relationship between communication media technology and human communication and to make more effective use of telecommunications technology in sharing and distributing cognitive models.

3.3 Conclusion

This chapter introduced key concepts, techniques, notations and tools in EM. Four illustrative case studies were considered in this introduction: the OLS regression to illustrate the use of definitive scripts in EM; the CAPM to illustrate the generalization of the spreadsheet concept in EM; the story of a retail trade in NYSE to illustrate the use of ISM and LSD notation; and the distributed stock market game to illustrate distributed EM. The distinctive qualities of model building in EM were highlighted. These include the focus on state as experienced; the maintenance of a semantic relationship between the EM model and its real world referent; experiential knowledge construction; and the use of definitive scripts in collaborative distributed modelling. These qualities give EM the potential to meet the technical and strategic demands for the wider agenda of computing that were introduced in chapter 2. This will be discussed in the following chapter.

Chapter 4

Empirical Modelling Technology For The Wider Agenda For Computing

4.0 Overview

Chapter 3 presented an overview of the characteristics and properties of model building in EM. This chapter considers the implications of EM on the technical and strategic demands for the wider agenda for computing in finance. Section 4.1 discusses how EM tackles the technical demands for this agenda. It argues that EM introduces a paradigm shift in the software system development activity by taking into account the experiential, the situated, and the human centred aspect of SSD. This section presents also the efforts of EM in researching a suitable framework for deploying prevalent technologies to support the paradigm shift in SSD. Section 4.2 attempts at illustrating, with diverse case studies, how the strategic demands for the wider agenda for computing in finance can be addressed by adopting the technical solutions provided by EM. The case studies provide a proof of concept of the potential success of EM in establishing a closer integration between the computing activity - the software system development activity - and various activities in the real world domain, that can be better mediated with computer-based technology. These activities include: communication, learning, business process modelling, decision support, product design, and visual exploration. The relevant case studies considered are: the railway accident, the virtual electronic laboratory, the warehouse, the timetable, and the attribute explorer. Section 4.3 concludes with potential prospects for the success of EM in tackling the wider agenda for computing in finance at the institutional, market, and investment levels and constitutes a background for case studies tackled in subsequent chapter 5, 6, 7, and 8.

4.1 Meeting the Technical Demands of the Wider Agenda for Computing in Finance

The technical demands for the wider agenda for computing in finance imply the need for greater integration between the human and automatic activities (semi-automation), the account for the situated and experiential aspects of the SSD activity, and the novel use of computer-based technology with other technologies (cf. electronic devices). This motivates a paradigm shift in the software system development activity and its corresponding computer-based support.

4.1.2 The Paradigm Shift in SSD

This section argues that the distinctive qualities of model building in EM introduce a paradigm shift in the software system development activity and its computer-based support: i. using an EM approach, modelling evolves as a central activity throughout the entire SSD activity; ii. the SSD activity in EM is characterised by its openness, situatedness, and the absence of the adoption of circumscribed methodologies that reduce the flexibility of the SSD activity; iii. users-developers-designers interact collaboratively in a situated manner to cultivate requirement engineering in the SSD activity.

i. Empirical Modelling Technology motivates a shift in emphasis in the software system development activity from programming to pervasive agent-oriented explanatory modelling.

Modelling vs programming in traditional SSD

Software is the means that enables a computer to become a tool for solving problems. The traditional software development cycle involves five stages: requirement analysis, specification, design, implementation, and verification. Modelling activity[1] takes place in the requirement engineering, analysis and design phases, whereas the programming activity is involved mainly in the implementation stage. These two activities are fundamentally different in character. This stems from the fact that the modelling activity is more closely related to the real world domain and involves more human insight and input, whereas the programming

[1] Using specification languages such as the Unified Modelling Language UML, and cooperating sequential processes CSP.

activity assumes a detachment from the real world and a reliance on abstraction and circumscription (cf. Figure 4.1).

Traditional Software System Development Cycle

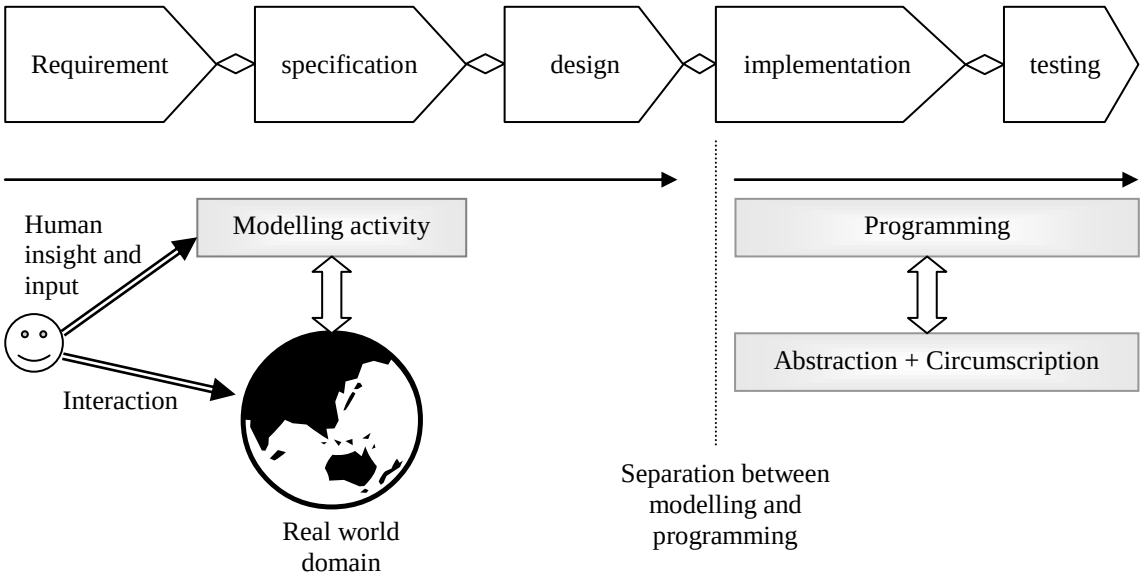


Figure 4.1 Modelling vs programming in traditional SSD

As software systems have become more complex, there has been greater emphasis on the role of modelling in SSD supported by new modelling approaches[2]. There has also been a complementary evolution in programming languages[3], whereby the implementation phase can

[2] Such as Agile Modeling AM (www.agilemodeling.com) that offers a practice-based methodology for effective modelling and documentation of software based systems [Amb02]. AM is a collection of values, principles, and practices for modelling software that can be applied on a software development project in an effective and light-weight manner. An agile modelling approach can be adopted in requirement, analysis, architecture, and design. However, AM is not a complete software process. AM focuses mainly on effective modelling and documentation. It does not include programming activities although it recommends to prove the model with code. AM addresses the issue of how to apply modelling techniques on a software project taking an agile approach such as extreme programming (www.extremeprogramming.org), Dynamic Systems, Development Methods (DSDM) (www.dsdm.org), or SCRUM (www.controlchaos.com).

[3] Programming languages can be classified by generations: Machine codes formed the first generation; auto-codes and assemblers formed the second generation; high-level languages such as Fortran, Cobol, and Pascal formed the third generation; languages for non-professional programmers, such as spreadsheets, are considered as fourth generation. Fourth generation languages include decision support languages. Programming languages can be also classified by their purposes: general multi-purpose, interactive, special purpose. Another classification would be by operation: the fundamental operation of functional languages (e.g. LISP) is the evaluation of expression; the fundamental operation of imperative languages is the use of sequence of commands (e.g. Fortran, Cobol, Pascal, Basic, C); declarative languages attempt to combine the operation of functional and imperative languages [WC88].

use higher level abstractions better suited to the application, thus offering greater support to SSD[4]. Neither of these developments has had a significant impact on the nature of the relationship between the modelling and the programming activities. Programming still presumes a preconceived and abstractly specified functionality and modelling is still used as a means to identify this preconceived functionality. To reach the programming stage the product of the modelling activity has to be rich enough to deliver an appropriate circumscription of the real world associated with a reliable pattern of behaviour and a specified functionality. This implies that modelling takes precedence over programming and that the starting point for the programming activity is determined by the outcome of the modelling.

EM for SSD

EM aims at reconciling the imbalance between theoretical (abstract) and empirical (real world) ingredients in current SSD practices [Hon99]. Empirical Modelling technology has evolved as a means for modelling application domains by focusing on the representation of the state of the real world mediated through metaphorical representations that are developed in a situated manner. The impact of this evolution has been to show that EM activity can lead to the construction of computer-based artefacts whose relationship to their real world context is very similar to that reached in traditional modelling for SSD at the point of circumscription[5]. There are then two significant differences between Empirical Modelling and traditional modelling for SSD:

(1) In EM, the computer-based artefact exists prior to what would serve in traditional modelling for SSD as a point of circumscription.

(2) In EM, the modelling activity can carry on beyond what would serve in traditional modelling for SSD as a point of circumscription, in effect taking the place of the programming activity.

In an EM approach to SSD, the artefact reflects the distinctive qualities identified in section 3.2. These give the artefact an explanatory and agent-oriented character that enables it to be integrated closely with activities external to the system.

---

[4] Extreme Programming (www.extremeprogramming.com) is an example of a disciplined approach to software development that witnessed success in the industry because it stresses testing, customer satisfaction and team work, while implementing simple ways to enable groupware style development [Bec00]. XP aims at improving a software project in four essential ways: communication, simplicity, feedback, and courage.

[5] In particular, see the doctoral theses of Yung (1992), Ness (1997), and Sun (1999).

ii. EM technology proposes an open, situated, and amethodological approach to Software System Development (SSD).

SSD in theory and practice

Studies in the area of software system development focus on the process initiated when capturing the statement of the problem of a client and terminating at the delivery of the software product that will potentially solve the problem. Britton (1993) views system development as a gradual progression from the client's initial vague ideas about the problem, via a series of transitional stages to a complete formal statement, expressed in a programming language, which can be executed on a machine. A typical life cycle of a software system may be divided into seven stages: problem definition, feasibility study, analysis, system design, detailed design, implementation, and maintenance. The early phase of the software system development cycle is commonly referred to as software requirement analysis or software requirement specification [Dav93]. Different approaches, techniques, and notations are used to model a system. Some approaches involve constructing many different models of the system - each leading to a partial description of the system. On account of this, understanding the whole system necessitates an understanding of the relationship between the different partial models and their complementary role. In current software system development practices, there is a clear separation between the representations used in specifying and implementing the developed product. The diversity of applications and the rapidly changing environment pose great challenges to the software system development process. Adopting a general, context independent, methodology to structure the software development process and all its phases is a challenging task because different applications require different approaches. Britton (1993) argues that it is impossible for a single development methodology to prescribe how to tackle the great variety of tasks and situations encountered, and proposes the development of a toolbox of techniques and skills, modelling tools, and approaches to support the development process. Adapting the software development process to a rapidly changing environment is a difficult task especially for ill-defined and volatile software systems [Flo87].

SSD in EM

EM technology proposes an amethodological approach to Software System Development [Sun99]. The proposed software system development process consists of a collection of situated activities that arise in the construction and use of the required system in the real world. This puts a greater focus on the interaction between human agents involved in the software development process and the product-under-development. The product under development is represented by an interactive artefact (ISM) that reflects the evolving software system. The proposed amethodological approach must take both technical and social aspects

into consideration in the software development process. In EM, Software System Development begins with an open ended analysis of a domain [RCR00] that informs the development of a system with dynamic rather than rigid boundaries. Deriving useful systems from an EM model necessitates an appropriate circumscription of the model once a desired functionality can be relied upon. However, the boundary of the system need not be preconceived, but can grow as the modellers' understanding develops [RCR00].

iii. An EM approach considers SSD activity as a human centred activity involving a social network of users-developers-designers interacting collaboratively to cultivate the requirements for the system under development

Users and developers role in SSD

Historically, the view of SSD has been transformed from a technical to a social one [HKL95], as technical processes are unable to cope with all the issues[6] surrounding the SSD activity [HKL95, CS90, Mum95]. Many approaches that regard SSD as a social activity have been introduced [CS90, AF95, Mum95]. Their key concern is to facilitate the interaction between the users and developers in the SSD activity. The professional work practices (PWP) approach [Kyn91] aims at improving developer's professional work practices. The co-operative design approach to SSD puts the emphasis on users' involvement as developers. It argues that technology should encourage reciprocal learning between users and developers [CWG93]. Neither PWP nor co-operative design is a method-based approach; they provide principles, concepts and techniques to support a social perspective in SSD.

User-developer-designer collaboration in an EM approach to SSD

SSD in EM involves building a computer-based artefact reflecting the views of users and developers (designers) [Sun99, BRSW98]. This shifts the emphasis from a developer centred approach to SSD that abstracts user requirements rigidly and detaches these requirements from their real world context. In contrast to existing SSD practices that eliminate the human factor from the SSD process by adopting rigid algorithms and preconceived mechanisms, EM regards SSD as a social process that needs technical support. This technical support goes beyond the use of computer for knowledge representation to knowledge construction and knowledge sharing. Distributed Empirical Modelling provides a framework for the collaboration of the users and developers (designers) to cultivate the requirements[7] of the

[6] These include job satisfaction, user resistance, learning and interpersonal interaction.

[7] The early phase of the conventional software system development life cycle is commonly called software requirements analysis or software requirements specification [Dav93]. It involves analysing the software problem and concluding with a specification of the desired external behaviour of the

system under development. In the Distributed Empirical Modelling framework (DEM), the interpersonal interaction between users and developers is supported at two levels: modellers, as external observers, can intervene to shape agency in a concurrent environment; or modellers, as internal obervers, act as agents to carry out interaction through pretend play.

Requirements engineering in an EM approach to SSD

In EM, the situated and context-dependent character of the requirements development phase is emphasized. A framework entitled Situated Process for Requirements Engineering [Sun99] is established, where human agents interact within a collaborative environment, on the basis of their current context and resources, to identify requirements [CRB00]. From an EM perspective it is difficult to limit the requirement engineering activity to a single stage in software system development. This activity is conducted gradually and pervades the whole of the software system development. This is a promising but technically challenging approach to SSD. The 'opportunity cost' of meeting evolving user needs is not being able to come up eventually with a finished product.

4.1.2 The Broad Foundation For Computing

It is unrealistic to expect Empirical Modelling technology alone to lead to a paradigm shift in SSD. The support of other current technologies (artificial intelligence, databases, and virtual reality) is needed. EM technology aims at identifying an appropriate framework for deploying these technologies to effect a paradigm shift in SSD. This has been reflected in EM research directed at: i. new foundations for AI; ii. a new perspective on database technology; and iii. principles for wider use of VR technology in modelling social domains.

i. EM technology proposes new foundations for artificial intelligence that motivate a shift from a logicist AI approach to an experience based AI approach where the activity of developing construals to represent experiential knowledge of the real world domain is central.

The experiential dimension in AI

Broadly speaking, AI is concerned with creating computer programs that can emulate human decision making and behaviour. In approaching AI, Empirical Modelling attempts to find a better ontology for the terms intelligence, learning, artificial intelligence, metaphors, and agent. It addresses the criticism faced by the logicist approach to AI, and investigates the

software system to be built. This analysis is referred to as functional description, or functional requirements. Problem analysis and product description are two types of activities conducted in the conventional software requirements phase.

principles and foundations of a computational framework for artificial intelligence that account for private experience in the real world. It also establishes a relationship between language and learning [Bey99].

Private and public knowledge in AI

Empirical Modelling Technology relates the concept of learning and intelligence to the migration from private experience to public knowledge and from empirical interaction in the world to theoretical formulation. This migration involves a series of sequential activities: the interaction with artefacts, the construction of practical knowledge, the identification of agency and dependency, the identification of generic patterns of interaction, non-verbal communication, the identification of common experience and subjective knowledge, and the symbolic representation and use of formal language to describe the private experience. An analogy can be drawn with the migration from private experience to the linguistic description of this experience. Human intelligence is related to the empirical mechanism that leads to the formulation of theories that formulate and circumscribe a reliably occurring pattern of experience. A logicist approach to AI [Kir91, LF91, LI95, Der87, CW96, Bro91, Bro91-2] attempts to frame human activity around objective knowledge and skills. It typically requires all significant operations to be based on highly reliable expectations, and assumes that practice is context independent. In contrast, EM technology accounts for subjective experience and knowledge and advocates an empirical approach to construing a system. A construal in EM is empirically established and experientially mediated. The choice of agents is pragmatic and the changes in the state of the system are not preconceived.

The EM frameworrk for deploying AI

The basic principles and foundation of the Empirical Modelling framework for AI are concerned with construing phenomena in terms of observable, dependency and agency, and constructing physical artefacts, mainly computer-based, to represent them. A virtual correspondence is established between the real world referent and the artefact through continuous interaction with the artefact and its real world referent. The EM approach to AI emphasizes the importance of accounting for subjective and private knowledge (first person concerns), provisional and unreliable insight, and the particular context under consideration. Unlike a closed world simulation, an EM artefact is open and subject to modification in the light of future experience.

In thinking about AI in EM, three perspectives are identified according to the roles for agency in the corresponding computer-based model and the extent to which its behaviour is circumscribed: a first person perspective (the corresponding computer-based artefact represents subjective private and personal knowledge); a second person perspective (the role of agency played in the artefact is projected from a first person to a second person); and a

third person perspective (the computer-based artefact resembles a computer program, where agents are predefined and their role is circumscribed and formally specified).

ii. In EM, a database is regarded as a computer model that enables its user to simulate different modes of interaction with a real world system.

The principles of agent oriented databases are elaborated in [BCY95]. Tools to support these principles are under development. From an EM perspective, a database can be regarded as a computer model that enables its user to simulate different modes of interaction with a real world system. Agent oriented modelling is adopted to develop the database computer model. A system of definitions (definitive scripts) is used to record the current values of real world observables, and generalise the principles adopted in the relational language ISBL[8]. EM thinking about agent oriented databases aims to confront the challenge of the changing nature of data and agency in modern database applications and the manner in which data is presented to and manipulated by the user. An EM model that illustrates data manipulation subsuming many traditional database functions is the timetable model discussed in [BWMRR00].

iii. Virtual Reality technology can potentially benefit from adopting EM principles and techniques in modelling complex domains. An EM - VR merge is proposed.

Ontological issues bearing on the questions: "What is reality? What is virtual reality? What is the correspondence between the two?" have been considered in EM [Bey99]. The major contribution of EM technology to VR technology is the proposal for a new technical approach to shape visualization embedded in internet web-browsers alternative to that provided by VRML. This approach, pioneered by Richard Cartwright [Car00, Car99], is based on the transmission of recipes for building shape models at the internet browser. It provides a high degree of open-ended interaction flexibility for users, compact transmission of models over networks, support for distributed multi-user shape modelling, and the representation of data dependencies between application domain data and shape models. Cartwright's concept of *Empirical Worlds* emerged in the course of applying EM principles to computer-based geometric modelling. The users of Empirical Worlds can generate new geometry on the fly without the need for the shape to have been preconceived and constructed as a set of polygons during the creation of a VRML world. A related proposal for exploiting EM technology in

[8] The relational database query language ISBL was an early example of a definitive notation with relational algebra as the underlying algebra.

deploying VR technology in a social application domain is introduced in [MBG01]. This proposal will be considered in more detail in chapter 6.

In seeking a suitable framework for deploying VR technology in social application domains, EM principles for human computer interaction HCI[9] are potentially of interest. EM research in open-ended HCI, as presented in [BRSW98], advocates a shift from user-centred design as introduced in [PRSBHC94] to the development of explanatory models where both the user and the designer operate within a computer-based environment whose state metaphorically represents the experiential context from which the domain knowledge is derived. Conventional HCI design detaches the designer from the real world environment, whereas HCI design in EM advocates user and designer collaboration and their interaction with the real world environment to develop an explanatory model that serves the development of the computing system.

4.2 Meeting the Strategic Demands of the Wider Agenda for Computing

4.2.1 Closer integration between the SSD activity and diverse activities in real world domains

The paradigm shift in SSD motivated by EM enables a closer integration between the SSD activity and activities undertaken in different application domains, with greater advantage and

[9] Human computer interaction (HCI) is inherently a multidisciplinary subject. It involves theories of human behaviour as well as the principles of computer systems design [PRSBHC94]. Exploration of the social and organizational context of the user is also important. Research in the area of HCI aims at developing the right system to fit a specific purpose, it involves studying users and their tasks and relating this information to design styles, human factors theories, guidelines and standards in order to select an appropriate form of interaction. Computer supported cooperative working (CSCW), hyper and multimedia and virtual reality are state of the art in applications in the area of HCI. It is commonly argued that future human-computer interaction will come to resemble our everyday interaction with the world [BRSW98]. Attempts are made to frame HCI in terms of three interdependent abstractions: methods, tools, users. This tends to promote the view of the computer as a general purpose tool capable of playing the role of many pre-existing tools such as calculator, folders, and desktop utilities. This is challenged by novel uses of computers (beyond conventional uses of existing tools) emerging with the advance of computing and communication. Preece (et al, 1994) believe that there is no single solution to the problem of how to do interaction design. Beynon (et al, 1998) argues that everyday experience of our environment is characterised by an openness to interact, and an ignorance of the consequences of our interaction, that has no counterpart in interaction with closed-world computer models.

potential for success in social application domains. Interpersonal communication, learning, participative business process modelling, strategic decision support, product design and visual exploration are all examples of activities in different application domains where computer-based support has failed so far to address its intended objective. The following points summarize the broad implications of EM technology on these activities.

- ***In computer-based interpersonal communication:*** *EM technology promises richer support for computer mediated interpersonal communication by accommodating various modes of interaction, providing distributed communication of definitive scripts, enabling collaborative communication, and attributing various agent-based roles and views to participants.*

- ***In computer-based group learning:*** *EM technology promises greater support for experiential group learning activity. Empirical Modelling Technology is concerned with representing the processes that lead to the discovery of concepts [Bey99].*

- ***In strategic decision support[10]:*** *EM technology proposes an experience based approach to strategic decision support where human and computer activities integrate more coherently in exploring a decision ([RR00], [BRR00-1], [BRR00-2], [RCR00]). EM technology can play three roles in SDSS: it helps in coping with imprecise, qualitative problems; it offers end-user involvement, and distributed access to users. The spreadsheet has been a primary tool for decision support, especially in conducting what-if scenarios. Empirical Models generalise the spreadsheet use in decision support in three respects: presentation, underlying algebra, and agency.*

[10] Decision making is a central human activity. Simon (1960) identifies three phases of decision making: intelligence, design, and choice. Mintzberg et al. (1976) conceives the process of decision making in three major components: problem identification, development of alternative solutions and selection among the alternatives. The computational paradigm that best matches this characterisation of decision making is that of logic programming [Kow79]. Decision making is classified in three levels [Sut89]: strategic, tactical, and operational. Strategic decision support is the most challenging as it involves qualitative description of the system, and decisions either to be taken by top management (end-users), and/or likely to be shared among managers (Group Decision Support Systems). Davis (1996) argues that decision support is more a service than a product. Decision support systems (DSS) are computer-based systems that bring together information from a variety of sources, assist in the organization and analysis of information, and facilitate the evaluation of assumptions underlying the use of specific models [Sau97].

- ***In visual exploration:*** *EM technology proposes an amethodological approach to data analysis and exploration that helps in revealing qualitative and hidden aspects of relationships between data.*

- ***In participative business process modelling (BPM)[11]:*** *EM technology proposes a framework that complements a structured approach[12] to business process modelling with an experience based approach to participative business process modelling where the role of the human in partaking in the modelling process is preserved. A shift in perspective from the paradigm of method-tool-user to a paradigm where the computer and human integrate more coherently to achieve the intended business objective is proposed. EM technology advocates a shift in perspective from specifying a business process by an abstract pattern to specifying a process as developed from a situated[13] activity. In seeking a reliable pattern in the business process, a semi-automated, situated style of modelling is proposed.*

- ***In participative business process re-engineering (BPR)[14]:*** *Recognizing the challenges faced by BPR in attempting to move from the problem domain (real world application) to the solution domain (world of programs and systems) [CRB00] and the difficulty in communicating between the business and IT specialist, Empirical Modelling Technology emphasises the importance of business process comprehension to enable process renewal. The business context and domain are important factors to be considered in BPR. EM*

[11] A business process is a series of activities to achieve a business goal. Modelling a business process means expressing the flow and the dependencies of steps in the respective processes in order to make the dynamic behaviour explicit, to be able to communicate it, to analyse it with respect to possibilities of improvement, and to use it for simulations as well as for controlling automated workflow [Sko98]. This definition embraces the meaning, the objective, and the use of BPM.

[12] Business management aspires to establish environments and routines that operate according to conventions and protocols as reliable as scientific laws [BRR00-2]. This aim might be difficult to achieve because a business process cannot be detached from its real world context as in the case of scientific and engineering applications. A business process is shaped by its environment. Many external, internal, and social factors affect the business process. Rules of the business process might be difficult to observe or adapt to.

[13] Situated models of business processes are motivated by the instability in the business environment.

[14] Re-engineering a process means discovering how a process currently operates, redesigning that process to improve efficiency and remove wastage, and finally implementing the new process using whatever technology is deemed appropriate. BPR seeks to devise new ways of oranizing tasks, organizing people, and making use of IT systems so that the resulting processes better meet the goals of the organization [VRWH94]. Re-engineering entails a radical change not only an improvement [Dav96, Dav96-2]. BPR activity embraces a wide range of disciplines [FRKBCJ00].

technology proposes a SPORE (Situated Process For Requirement Engineering) framework[15]*, encompassing participative BPR (i.e. supporting many users in a distributed environment) [CRB00].*

- ***In product design***[16]***:*** *Empirical Modelling technology proposes a modelling approach for product design based on observation and experimentation and lays a foundation for a computational environment for modelling a product in its situated context. A high level of interaction, openness, and good presentation media are major requirements identified for this computational environment.*

4.2.2 Illustrative Case Studies

The following review of case studies undertaken throughout the history of the Empirical Modelling projects reveals broad implications of EM on the integration of computer-based technology with real world activities that are inherently not computer-based.

The Railway Accident
Illustrating the EM implications on Computer-based Interpersonal Communication

Motivation

Human interaction within a social network is rich, context dependent, and situated [Son93], [SBK00], [Son99]. The relationship between communications media technology and human communication is key in determining the extent to which human communication is enhanced by computer mediation. Moreover group activity is essential to redress individual bias in subjective perception (cf. Gruber's and Sehl's Shadow Box experiment [Goo90]) and to

[15] The SPORE framework was proposed by Sun (1999) in his doctoral thesis [Sun99]. Within this framework, requirements are viewed as a solution to the problems identified in the application domain and are developed in an open-ended and situated manner. People participating in the requirements engineering process are able to cultivate requirements through collaborative interaction with each other. This collaboration aims at solving problems rather than formal specification of requirements. The input to the SPORE situated model are: central problems of the domain which are subjectively determined by participants; relevant contexts such as organizational goals and policy and the relationship between participants, and available resources such as documents, technology and past experience. The outputs of the model are provisional solutions, new context, new resources and new problems feeding the model. The model is iterative and incremental. This means that it is built from a sequence of structured development cycles.

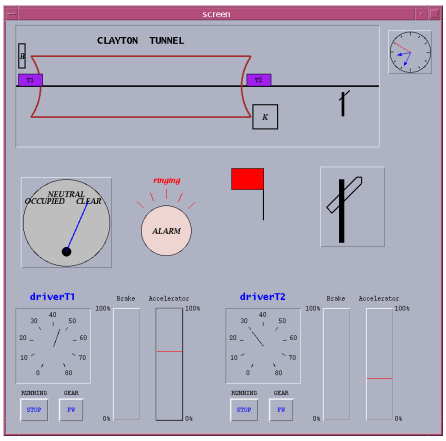
[16] Product design is the activity of generating a solution that meets the requirements imposed on a product [ESc93]. Fischer (et al, 2001) emphasize the importance of the designer creativity during the design process. This creativity is stimulated by the designer observation and knowledge about the product or its requirements. This knowledge is continuously growing as the design progresses.

share insight in cognitive activities including design [Son93], software system development [Sun99], decision support, learning, etc.. This motivates the consideration of a case study that reveals the complexity of interpersonal communication, its impact when mediated via communication technology, the various objectives that can be met through distributing the computer-based activity, and the practical application of emerging technologies for distributed computing.

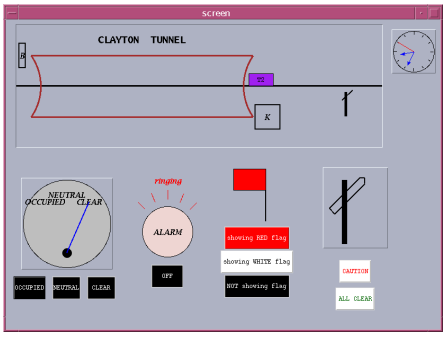
About the model

The railway accident model was developed by P. Sun as a practical illustration of the Distributed Empirical Modelling (DEM) Framework. It is regarded as one of the basic models to demonstrate the practical application of EM principles to various areas in computing. These include:

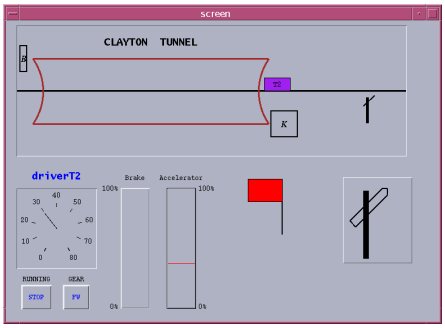
- user-developer-designer collaboration in the course of requirements engineering in software system development [SB99];
- understanding the impact of communication technology on human communication and proposing principles for computer mediated interpersonal interaction that support the sharing of cognitive models [BS99].



God's View (External Observer)



Killick's View (Internal Observer)



Driver's View (Internal Observer)

Figure 4.2 Snapshots of the railway accident model

The model aims at reconstructing a railway accident that occurred in the Clayton Tunnel near Brighton in 1861 [Rolt82]. The modelling has involved constructing computer-based artefacts to represent the views of the human agents involved in the accident (the drivers and signalmen as internal agents) and the global view of an external observer with exceptional state changing privileges (cf. the snapshots of the various views in the model). The model supports sharing insight into the individual understanding of the practices of the signalmen and drivers and how these contributed to the accident.

The personal agents' perspectives are specified by definitions that shape their agency of the use of different observables in the model. Attached to this definitive script is a visualization that represents the modeller's view and the roles of different agents in shaping the state of the distributed interaction.

Conclusions drawn from the model

The case study highlights several advantages that distributed modelling has over centralised modelling in general, as well as others that are specific to the use of EM. An open development paradigm, enabling various modes of interaction and giving various agency roles for participants in the modelling activity, supports a better understanding of the features of the model and the prospect for its use in conjunction with diverse activities in the real world domain. Identifying observables, agents roles, and the view of each participant in the modelling activity is essential for establishing the context of interaction among modellers.

In particular, the distributed railway model reveals the contribution of EM technology to distributed modelling in computer science in three respects:

1) *supporting agent roles and diverse modes and context of interaction:* In DEM, participants acts as internal agents in their own views and through their own privileges of action. A participant's view in DEM is different in character from an interface in conventional programming. It represents his / her provisional[17] working space including the metaphorical representation of observables that can be seen (**`oracles`**) and manipulated (**`handles`**) at any one moment in time.
2) *distributing the centralised spreadsheet modelling:* The communication of definitive scripts is revolutionary in extending the centralised spreadsheet modelling to accommodate various participants with diverse styles and privileges of action. It extends the openness of the spreadsheet modelling activity at the network level and supports it with views that metaphorically represent the working space of each participant in this modelling activity. The interaction of participants is not limited to features provided by their views but can be endlessly extended (within the attributed privileges) through the communication of definitive scripts. Such communication does not need to follow any preconceived pattern or sequence. In the railway model, the interaction between participants (drivers, signalmen) can be established at any instant through the communication of definitive scripts that affects the state of shared observables in the distributed views of the model (e.g. the communication of definitive scripts concerning

[17] The term 'provisional' indicates the possibility of change.

the observable **`clear position`** of the telegraph shared in the views of the two signalmen, such communication establishes the functionality of the telegraph device).

3) *supporting openness of interaction:* In DEM, the modelling activity can accommodate different scenarios and different construals for each scenario. The human role in leading the distributed modelling activity is central. Human participants collaboratively share insight and understanding to the extent that the computer-based interaction resembles the real world interaction with its openness and situatedness. In the distributed railway model, different scenarios can be played by human participants (e.g. the accident scenario, and the normal railway operation scenario). Also, a scenario might be construed and played by human participants in different ways (e.g. it might be the case that the inattention of the driver that was the main reason behind the accident, or the carelessness of one of the signalmen in operating the telegraph that lead to the catastrophe). Although the visual representation in the distributed EM railway model is unsophisticated in visual effect, the openness in interaction is different in character and far richer from what can be supplied by a VR scene of the accident where the number of scenarios and the construals of each scenario are preconceived.

The Virtual Electronic Laboratory
Illustrating the EM implications on Computer-based group learning

Motivation

Computer-based technology promises to play an important role in educating the future generation. However, no one can predict to what extent the computer will replace the teacher, the pencil and the book in modern schools and universities. Distance and group learning are emerging activities that tend to rely on computer-based technology and communication. These kinds of approaches to learning have a long history. There exist different learning theories that focus on observation and experimentation as an important aspect of learning. These include: constructivism, behaviourism, Piaget's Development Theory, Neuroscience, Brain-based learning, learning styles, multiple intelligences, right brain / left brain thinking, communities of practice, control

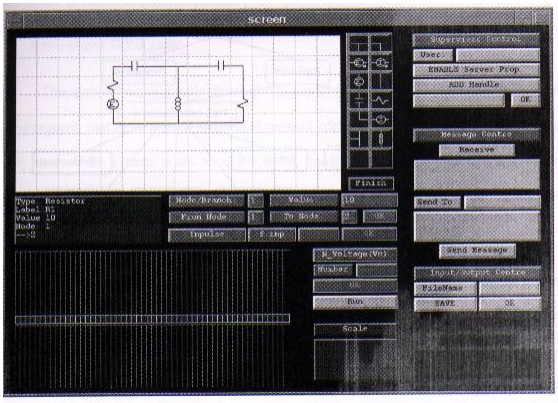


Figure 4.3 The Teacher's View in the VEL Model

theory, observational learning, Vygotsky and social cognition, problem-based learning, etc.. EM research [Roe99, COMICAL[18]] has directed special attention to providing a suitable computational framework that can support the implementation of different experience-based learning theories. Key EM concepts of observables, agency, dependency, the focus on state as experienced, the semantic relationship that is established between a computer-based model and its real world referent, and the use of definitive scripts in distributed modelling, give greater potential for EM to integrate the computer-based activity with the experiential group learning activity. EM computer-based artefacts reflect the modeller's mental model and his personal insight and understanding. This artefact is constructed based on past experience and can be amended in the light of newly gained experience and hence can support learning by experience.

About the model

The virtual electronic laboratory (VEL), developed by D'Ornellas [Dor98] and Sheth [She98], is a distributed model of an electrical laboratory used for educational purposes. The model offers similar features to those offered by other educational software for electronics (such as PSpice), and added value features in distributing the modelling activity and giving the teacher more control over the students' activity[19]. The participants in the model are the students and the teacher. The teacher acts as an external and internal agent who can create and shape the context of interaction between students and their views, and assign to students diverse agent roles and privileges to change the state of the model. The student's view is a restriction of the teacher's view. This enables the teacher to intervene as a super agent in the model. Interaction with the model is open ended. This is enabled through the visual interface associated with participants' views of the model (student or teacher) and / or the tkeden input window. Communication can be established both between students & teachers, and between students & students. Given the distributed client server configuration, student-student communication at

[18] A project that aimed at introducing the concepts and practical use of EM tools to teachers in UK and to gain feedback on the potential use of Empirical Modelling models in schools in UK.

[19] Sheth (1998) identified four scenarios in a real life electronic laboratory: (1) the teacher and the lab assistant initially set up the equipment and the circuit on the circuit board containing all the required components. The students crowd around the table where the teacher is in control of the experiment. The teacher describes the circuit and its components to students and then asks students to change values of components for learning purposes; (2) each of the students either in-group of one or more create their own circuit on their own circuit board and monitor the outcome of their experiment. The teacher may assist students in building their own circuit. In this case the teacher does not have control over all the students who are working and can supervise only one group at a time; (3) the teacher may carry out the experiment on the computer and the students observe and ask questions. The teacher will demonstrate to students the effect of changing components values and the students learn from this interaction. This method is not based on a distributed architecture; (4) The teacher uses a software tool such as PSpice in aid of the experiment. Once this is done, the student will interact with the tool PSpice and make

the clients should pass via the server. Agency is assigned by the teacher to the student, to observe (oracle) or change (handle) the values and parameters of the electronic circuit components and its design, using the LSD[20] notation for distributed communication [Sun99].

Conclusions drawn from the model

The model highlights added value that can be gained from using EM as a framework for computer-based learning. Specific features in the VEL model to which EM makes an essential contribution are:

- The extended teacher's computer-based control over the student activity in the distributed modelling environment. In various existing electronic lab scenarios the teacher has either full computer-based control over the modelling activity with little computer-based control and intervention from the student, or she/ he can have minor computer-based control over the student activity.
- The openness of the interaction with the distributed model. This is facilitated through the interface for communication, and the open-ended communication of definitive scripts that is radically different from traditional means of distributed modelling.
- The situatedness of the interaction with the distributed model. Interpersonal communication that can take place between the teacher and students is not preconceived. It reflects modeller's insights and experiential learning activity.

In general, EM technology can support diverse modes of interaction in computer-based interpersonal communication. Some built-in interaction modes are provided by the tools (normal, private, broadcast, and the interference modes). However, different configurations for communications and modes of interaction can be established, through using the LSD notation in dtkeden, to suit various needs of computer-based interpersonal interaction. In a group learning context, EM technology can potentially deliver computer-based support that *integrates* with, rather than *substitutes* for, the human role in the learning activity. This promises a greater role for the computer as a resource used by the teacher and the student in modern computer-based learning.

The Timetable
Illustrating the EM implications on strategic decision support

Motivation

observations and learn. The teacher moves round the classroom giving attention to each student for a short time. The teacher has no control over what the students are doing directly.

[20] This notation is similar in syntax to the one introduced in chapter 3, but can be used only in distributed Empirical Modelling to set the context of interaction for different communicating agents.

Providing computer-based support for the human decision making activity is a challenging task. Various tools[21] and technologies[22] have been explored for this application. Despite the enormous advances in storage, speed, functionality, and interface design, the effectiveness of the computer-based decision support is relatively modest. This is mainly attributed to the challenges posed by the coherent integration of the human and automated activity (semi-automation) and supplying the computer-based support for the human cognitive activity.

EM technology seeks a computer-based framework that supports the continuous human engagement in the automated decision making activity, and the exploration of a software development paradigm that ensures a distinctive quality of interaction between the human and automated activity.

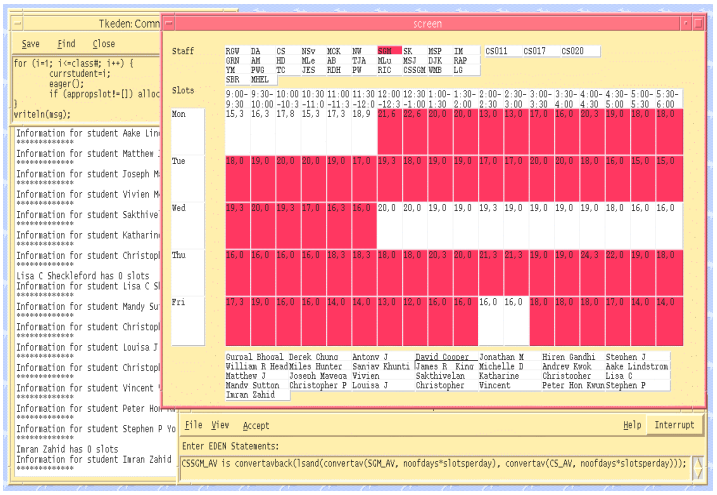


Figure 4.4 A snapshot of the Temposcope

About the model

A real timetabling application, developed by M. Beynon, S. Yung, S. Maad, A. Wong, and A. Ward is considered in [BRR00-1, BRR00-2, BWMWRR00] as a case study in computer-based decision support, the integration of human and automated activities, and the development of business solutions. It involves timetabling a week of oral presentations for final-year project students at Warwick. Each presentation requires a timeslot of 30 minutes duration between 9 am and 5 pm from Monday to Friday. The presentation is attended by the project supervisor, the second assessor, and a moderator. Staff are assigned to be moderators or assessors according to their suitability, availability, and workload criteria. The EM model developed is intended to provide semi-automated decision support for the human timetabler in recording and assessing the observables that are of crucial interest in developing the timetable. The model is similar to an instrument[23] in its use and development. The point of similarity to an instrument in its use stems from the scope for access and manipulation of *observables* of interest. The phases of the development of the timetabling instrument are: (1)

[21] Spreadsheets, databases, expert systems, neural networks, intelligent agents
[22] e.g. artificial intelligence, hypermedia, networks, visual programming
[23] The term Temposcope (**T**imetabling **EM** **P**rojects **O**rals Instrument) is intended to suggest the use of the computer as an instrument.

an assembly phase where the component parts individually tested on artificial data sets are combined to create the first prototype; (2) a testing phase on real data generated from an old timetable; and (3) a usage phase where the instrument is used in its intended mode (semi-automated or fully automated).

The semi-automated timetabling mode involves a high degree of interpretation and intervention from the human timetabler. Dependencies established between observables enable the semi-automation and the gradual construction of the timetable. A semantic relationship is established between the state of the model and its real counterpart (a working timetable satisfying all needs and constraints and accommodating unforeseen changes).

Supporting the decision making of the timetabler is much broader then deciding on the allocation of resources subject to well formulated constraints. Making human judgement, considering qualitative constraints, resolving singular situations are examples of timetabling activities that are hard to address with fully automated decision support.

Conclusions drawn from the model

Providing computer-based decision support relies on a high quality of integration of human judgement and automated processing in the decision making activity and a greater emphasis on the experiential aspect of a computer state. The open-ended engagement in the modelling activity and the semantic relationship between the state of the model and its real counterpart are distinctive qualities of model building in EM that better serve the decision support objective. The EM approach allows the entire timetabling activity to develop in a less constrained and less process-driven manner. It makes use of representations of state and agency that reflect the designer's construal and exploits these in both manual and automated aspects. Normally, the automation of the timetabling process exploits optimized representations of state that are difficult for the human interpreter to understand.

The Attribute Explorer
Illustrating the EM implications on visual exploration

Motivation

Real data with qualitative and imprecise characteristics are commonplace in many domains. Providing computer-based support for qualitatively richer exploration of data motivates a paradigm shift from conventional computer-based support for data analysis. This paradigm shift promotes:

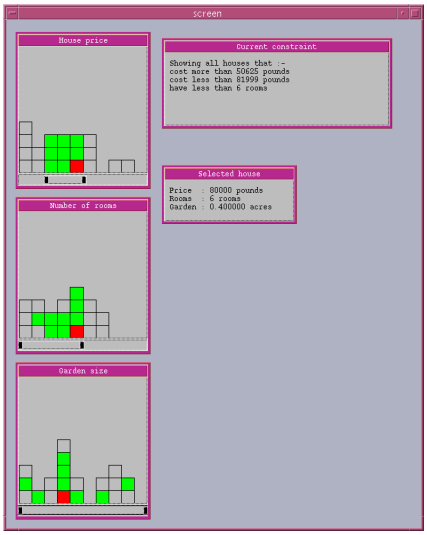
- visualisation in support of cognition (e.g. to complement abstract data processing);
- interactive exploration (e.g. to provide an alternative to preconceived search strategies subject to preconceived constraints);

- the discovery of hidden complex relationships between data (e.g. to complement computationally intensive analysis of tabulated data);
- the representation of qualitative relationships between data.

About the model

The EM Attribute Explorer, a data analysis model for definition driven interactive exploration of data is developed by C. Roe, firstly as an EM definitive script, then an appropriate visualisation is coupled to this script [Roe00]. The model is inspired from the Attribute Explorer (originally developed by Bob Spence, Imperial College London, and currently being further developed by him in collaboration with IBM Ease of Use Group). The model aims at offering an interface between the human and a data set. This interface acts as a lens to the data that can be looked at in different ways to cultivate our subjective understanding of the meaning of the data viewed. The definitive model can be customized to suit many different types of data (such as restaurant data, houses data, students marks, and weather records data). The first snapshot shows the visualisation of house data satisfying various constraints expressed in definitive scripts. The second snapshot shows the addition of non-numerical types (text fields, picture fields, map fields, etc.) to the visualisation of data about houses. This gives the modeller more immediate insight into the significance of the data in the real world.

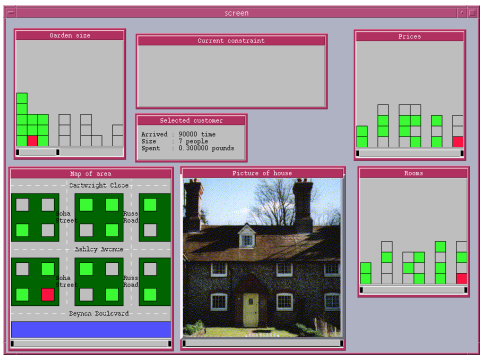
Figure 4.5 The EM Attribute Explorer



Visualisation of houses data satisfying various constraints

Conclusions drawn from the model

Conventional approaches to data analysis face many limitations. This motivates a paradigm shift in approaches for data analysis. Interactive visual exploration is a potential viable approach to data analysis that takes into account quantitative and qualitative features of data. EM principles, based on the concepts of observable dependency and agency for modelling personal subjective and provisional view of a domain, can potentially provide computer-based support to interactive visual exploration. The original Attribute Explorer is not a product of the EM group, but it clearly exploits dependency in a



Non-numerical types added to the interactive visual exploration

powerful manner. This has been demonstrated by C. Roe's re-implementation of the Attribute Explorer using a simple script. The Attribute Explorer epitomises one way in which EM concepts can potentially be exploited in visualization (other EM research along these lines is found in R. Cartwright's JAM and JAM2[24] tools and the Empirical World). There is a potential power in future use of the EM Attribute Explorer model. This stems from scope to link such exploration of data to pre-existing scripts.

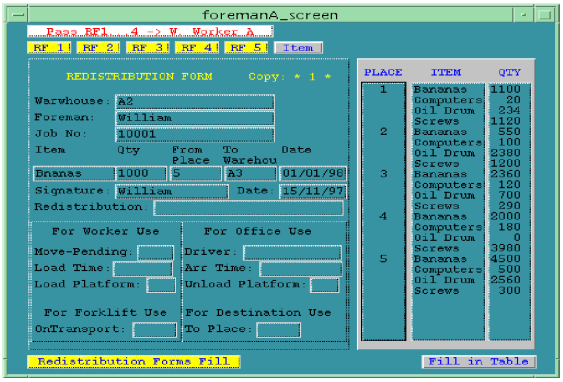
The Warehouse
Illustrating the EM implications on participative BPM /BPR

Motivation

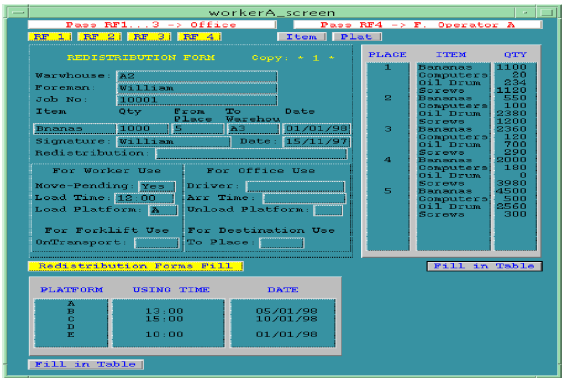
Applying a computational framework to business process modelling is a challenging task. This is widely acknowledged in software system development and software engineering practices and is mainly due to the complexity of business processes and the difficulty of understanding these processes and expressing this understanding in a model [CRB00]. Despite the aspiration of business management to establish environments and routines that operate according to conventions and protocols as reliable as scientific laws, adopting abstract methodological approaches to BPM leads to major problems [BRR00-2]. These include:

- enhancing the model incrementally in-line with further experiential knowledge of the business domain
- adding contextual information appropriate to different situations.

Figure 4.6 Views of different Human Participants in the Warehouse ISM



Foreman Artefact

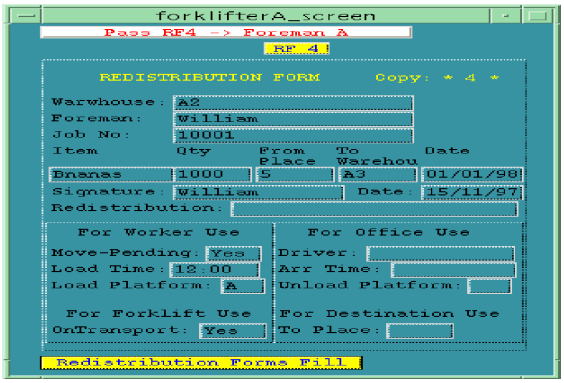


Warehouse Worker Artefact

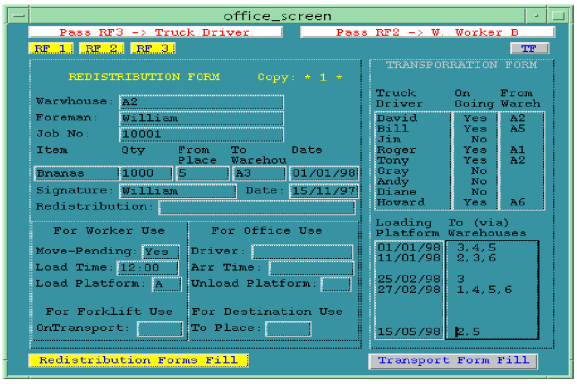
[24] /dcs/acad/wmb/public/projects/notations/JaM2API/DOCS/JaM2

- supporting end-user participation in the modelling activity.

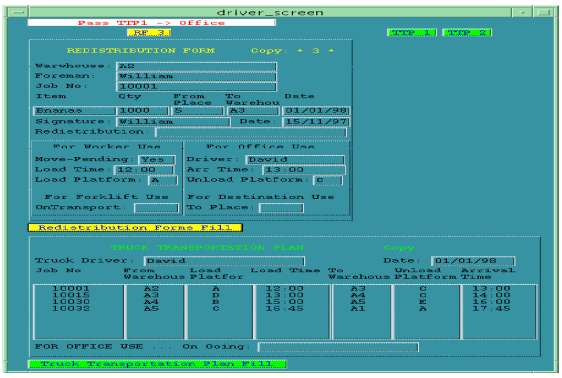
The limitations of abstract methodological approaches to modelling business processes motivate new modelling approaches that take into consideration the larger context of business processes (including the objectives of an organization, the viewpoints of people concerned in any particular process, the motives, the knowledge and expectations of users of systems being considered).



Forklift Operator Artefact



Office Artefact



Truck Driver Artefact

About the model

The warehouse[25] model [CRB00], developed by Y. C. Ch'en, is taken as a case study to illustrate the potential application of SPORE (Situated Process of Requirement Engineering) framework[26] in BPR. The case study was adopted by Jacobson et al (1992). The business process engineering involves introducing computer systems into the warehouse to offer automatic support to the storage and redistribution services. People in the warehouse include: the foreman responsible for the warehouse, the warehouse worker responsible for loading and unloading, the forklift operator

[25] The main function of a warehouse is to provide its customers with warehouse space. The operations of the warehouse include storing different kinds of items and using trucks to redistribute items.

[26] a problem oriented framework (based onn EM principles) in which requirements – viewed as solutions to the problems in the application domain - are developed in an open-ended situated manner [CRB00].

who drives the forklift in the warehouse, the truck driver who drives a truck between different warehouses, the office personnel who receive orders and requests from customers. An interactive situation model ISM is developed to analyse the business processes that operate in the warehouse application in order to re-engineer them. A key issue is to establish and maintain a semantic relationship between the ISM and its external referent. This involves explaining activities in terms of actual interaction amongst agents, then seeking to organize these into patterns that are reproducible and reliable. The modelling activity is based around patterns of information exchange (related to receiving, storing, and retrieving items from the warehouse) between warehouse personnel. Distributed modelling in developing the ISM makes it possible to separate the viewpoints of the agents in the model and to complement these with an external observer's interpretation. The above snapshots show the views of different human agents involved in the warehouse system.

Conclusions drawn from the model

Considering the wider context of business process modelling is a challenging task for conventional method-based modelling approaches. Empirical Modelling can potentially provide computer-based support for business process modelling in its wider context by adopting principles that aim at establishing a semantic relationship between the real world system and the corresponding computer-based artefact. EM technology emphasizes the human role in computer-based distributed business process modelling. Each participant in the business process modelling activity (human or human playing the role of devices) interacts with the distributed business process model in a situated manner through their own views and privileges for state changing actions. In the modelling activity, this reduces the burden of automating various roles of participants in a preconceived way, gives more room for personal insight, and can potentially support the detection of singular situations that might arise in the business activity when adopting a particular business process model.

4.3 Conclusion and Future Prospects of EM technology in Finance

This chapter has argued that EM technology can address the technical demands for the wider agenda for computing in finance by introducing a paradigm shift in the software system development activity. The EM approach to SSD takes into account the experiential, the

situated, and the human centred aspect of this activity. In addressing the technical demands for the wider agenda for computing in finance, EM technology seeks a suitable framework for deploying current technologies to support the needed paradigm shift at the computational level. EM technology proposes new foundations for artificial intelligence, the support of the application of virtual reality technology in different domains (graphics, and social domains), and proposes a new view for database technology.

This chapter has overviewed different case studies that show the added value from using EM technology in addressing the wider agenda for computing. It asserts the claim of the potential prospect of EM technology in supporting the following shifts in perspective:

In software system development practices

- from programming to modelling [Yun92]
- from a methodological approach to an amethodological approach to software system development [Sun99]
- from a closed to an open development model for software systems [Sun99]
- from a closed-boundary system development to dynamic-boundary system development [RCR00]
- from independent designer, developer, user activity to collaborative designer, developer, user activity

In the modelling activity

- from formal specification of behaviour to experiential representation of state.
- from separation between the experience of the real world and of the model to establishing a semantic relationship between the computer-based model and its real world referent [Sun99, BRR00-1, BRR00-2]
- from modelling preconceived activities to modelling situated activities [Sun99]
- from knowledge representation to knowledge construction
- from individual modelling to distributed modelling [Sun99]

In computer uses

- from modelling products in isolation to modelling products in a situated context [FB01, FB01-2]
- from using the computer as a tool to using the computer as an instrument

In computer-based support to various activities in the real world

- from business processes specified by an abstract pattern to business processes developed from a situated activity [BRR00-2]
- from a "method-tool-user" culture to a "human computing" culture where human and computer agencies are more closely integrated [BRR00-2].
- from a methodological approach to decision support (based on problem identification, development of alternative solutions, selection of a solution), to an experience based approach to decision support (based on situated problem solving) [CRB00], [BRR00-3].

EM seeks to provide an approach that enables a human agent to engage in situated activity and aspires at using the computer as an instrument rather than a tool. A situation is construed in terms of the concepts of: observables, dependency, agency, and agent.

EM tools are still basically research tools, they illustrate the principles of the technology but do not have the robustness and consistency required for a commercial product [RCR00]. However, this doesn't undermine the potential of the EM tools to test the foundations and practical applications in various domains.

In the author's opinion, Empirical Modelling technology will be seen in the future more focused towards domains where:

- Group social activity is important. Examples include collaboration in engineering design; user/designer/developer collaboration in the software system development process in all its stages; b2b (business to business) collaboration; collaborative product design in manufacturing; group decision support; participative business process modelling and re-engineering; and collaborative financial market modelling.
- The human role is central. Empirical Modelling technology will find greater support to applications in the area of social science studies where the human role is central in guiding the analysis exercising control over the system state.
- Knowledge is weakly structured: this is typical in the early stages of exploring a new field of study or initiating a system development process.
- The machine cannot replace human thinking and human machine integration is more appropriate.
- Novel uses of the computer are demanded.

The paradigm shift supported by EM and the distinctive qualities of model building in EM motivates the exploration of the application of EM technology in the finance domain at the institutional, market, and investment levels. This is considered in subsequent chapters with reference to selected case studies.

Chapter 5

EM For Integration and Virtual Collaboration In the Financial Enterprise

5.0 Overview

This chapter[1] establishes a framework for the application of the Empirical Modelling technology in the financial enterprise. Two applications in the area of software engineering and software system development are considered: ***software integration*** and ***virtual collaboration***. Section 5.1 considers the problem of software integration in the financial enterprise. Section 5.1.1 motivates financial software integration with specific reference to a challenging exercise involving the integration of ERP and e-commerce applications. Section 5.1.2 discusses issues relating to financial software integration and overviews the current approaches to tackling these issues. Section 5.1.3 frames the challenges for financial software integration and proposes new principles based on the EM approach to address them. A Situated Integration Model is proposed to meet the software system development agenda for financial software integration. Section 5.2 considers virtual collaboration in the financial enterprise. Section 5.2.1 overviews various forms of collaboration. The importance of taking human information behaviour into account in virtual collaboration is discussed in section 5.2.2. Section 5.2.3 considers the challenges to virtual collaboration. Section 5.2.4 proposes new principles for virtual collaboration based on the EM approach and illustrates these with reference to the case study of online trading. Section 5.3 concludes with the future prospects of EM Technology in the financial enterprise and speculates on the theme of building a web-based environment for corporate intelligent networks.

[1] This chapter is an expanded version of the combination of two joint papers by the author and Meurig Beynon [BM99, BM00].

5.1 Software Integration In The Financial Enterprise

This section addresses an important problem facing almost all financial institutions re-engineering their processes in the face of global competition pressures: the integration of existing technologies used in routine daily activities and in intelligent decision making.

Whereas there is a massive literature on the development of standalone software system applications solving problems in the real world domain, relatively few references can be found on the basic principles and practical applications of the integration of different standalone software systems. This represents a wide and deep gap in software system development and software engineering studies. ***Software integration*** embraces both ***technical*** and ***social*** aspects. The technical aspects relate to data and operations on data. The social aspects relate to the mode of interaction of human agents accessing the data and the context and environment within which this interaction takes place. Key issues for financial software integration to be discussed in this chapter include: a clear definition of the term "*integration*" and its scope and use in software system development; basic issues invoked in the integration; and techniques to solve the problem of integration.

Financial software integration is a situated context dependent activity. Considering a specific practical exercise in software system integration is only of limited benefit in identifying a methodology for software integration. Empirical Modelling technology proposes a framework for financial software system integration based on basic principles and techniques that inform the requirements engineering for software system integration. It promises to deliver two models: a situated integration model (SIM) to be used in the requirement engineering phase of the software integration process that combines conventional approaches to software analysis with the EM approach to software development and program comprehension; and a model of an integrated system accounting for views and agents within the system. The SIM helps in exploring possible modes of interaction, developing algorithms for integration, analysing distributed and shared access to data, combining visual interfaces, synchronizing the data flow, and providing a unified functionality.

5.1.1 The need for integration

Understanding, analyzing, and constructing experimental models of real life financial systems is a wide-ranging task that requires an integration of different technologies and enabling tools and calls for a bridging of the gap between theory and application as well as research and development in this area. Business process modelling, intelligent state and agent-oriented modelling, data warehousing and data quality assessment tools, financial analysis tools, and client server technologies should tie up coherently to enhance knowledge acquisition in a global financial market. Players in the global financial marketplace, such as investors, rating agencies, financial service providers, analysts and consultants, are faced with a massive amount of explicit and implicit market information characterized by a high level of dependency and interrelationship. An automated environment which, as far as possible, depicts and captures all aspects of the real-life financial system, is needed to adapt to state changes in global financial markets. Such an environment should provide a flexible human computer interface backed with a high level of interaction, visualization and reporting capability, with minimal overhead coding requirements.

Financial institutions are amongst the largest investors in computer technology and therefore appreciate the importance and significance of such technologies for their growth and survival. Financial institutions have often exploited technologies to create innovative products and services, capture market niches, and better serve the customer. The use of IT in the business and financial sector has evolved from the simple electronic data storage and limited computational capability of large business and financial databases to electronic data analysis and interchange. In the process, new forms of electronic data storage have been introduced, so that relational database management systems have been complemented by object-oriented databases and hyper-based storage. “Turning information into knowledge” is a major corporate challenge and the rate at which organizations learn and accumulate knowledge may become the only sustainable competitive advantage. With the growing complexity and competition in the global markets, the effective use of scarce resources and new technologies is of strategic importance. Enabling tools and technologies, such as business process modelling, intelligent analysis, data warehousing, and web technology should integrate coherently to establish corporate intelligence networks for intelligent information gathering, dissemination, and decision making. This intelligence network is critical for enabling global knowledge and information consolidation and distributing heterogeneous data relating to local markets (cf. Figure 5.1).

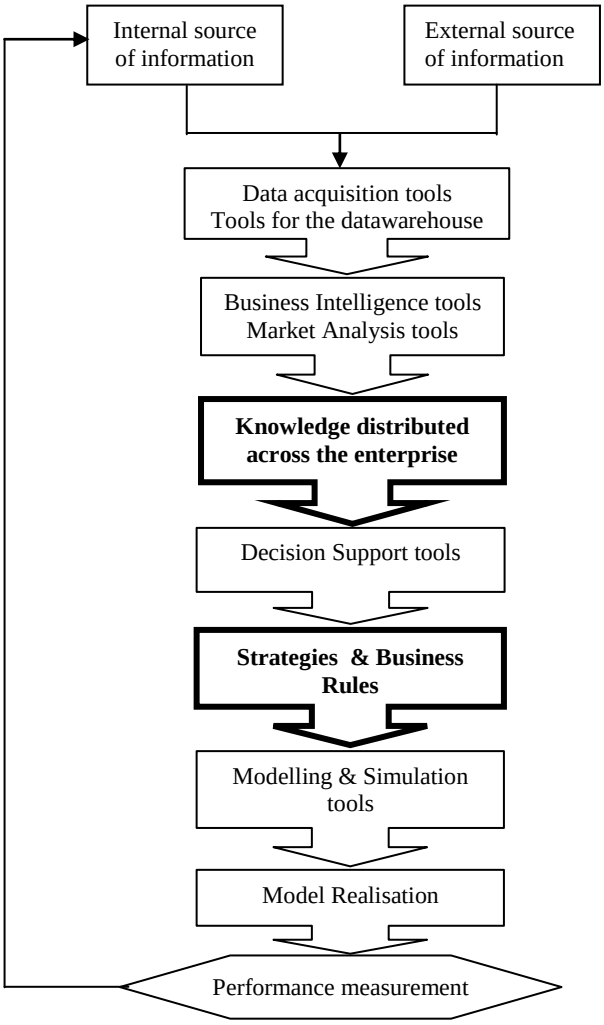


Figure 5.1 The integrated application chain

The urgent need to research various issues related to financial software integration is highlighted by considering the case study of the integration of: i. e-commerce applications with ii. enterprise resource planning[2] (ERP) applications.

[2] Enterprise resource planning (ERP) is an umbrella term for all tools and technologies used to handle the internal operations of a firm and to automate its business processes (ERP applications include controlling payroll, inventory, purchases, finance, personnel operations, etc.). The term *back office* refers to the IT centre where all enterprise resource planning applications are handled. ERP software applications used to run on mainframes, however, the advent of the year 2000 problem, and the introduction of the euro currency has forced many business firms and financial institutions to upgrade the tools and technologies used in their back-offices and to re-engineer their internal processes. Object Oriented technology is adopted in the design of many of today's ERP applications.

i. The emergence of the internet has challenged traditional business models through its ability to offer direct routes to market, to reduce barriers to entry and to increase the efficiency of trade activity [TW99]. Electronic commerce has been experiencing explosive growth[3]. Firms started their electronic commerce activity by first establishing a web presence, then by promoting some of their products and services online. These initiatives were not so expensive, they generated adequate profits, and increased the firm's interest in e-commerce activity. Today, electronic shops[4] are becoming more advanced, their design is more complicated, and their rewards and costs are higher. E-commerce applications have been developed with a view to establishing customised online stores with various design and catalogue structures. The workflow of transactions between the suppliers and buyers is becoming more efficient and less error prone when handled electronically. In a competitive global market, shops and trading firms selling their products and services online are facing a major question: how useful, profitable, intelligent and attractive is an online trading store?

ii. After a long period of sustained growth, the market for enterprise resource planning applications reached a saturation level and started its downturn. This is attributed to many factors including the diversion of funding to fix the year 2000 problem, the saturation of the ERP market, the high cost of acquiring and maintaining ERP applications, and the difficulty in introducing changes to these applications due to the inflexibility of their design and structure.To restore their viability and to tap into new

---

[3] Results of a recent survey from Visa International confirm that commercial electronic commerce is set to grow dramatically in Europe over the next five years, at a rate of about 30 times the growth of GDP in most European countries. Another e-commerce survey, conducted by PricewaterhouseCoopers in June 1998, predicts that within the next three years the major growth areas are extranets for business partners, company-wide data warehouses, customer service via the internet, web catalogues, and web-based transactions.

[4] A typical virtual shop model includes: (1) The *contact information* that provides address details and shopping support contacts. It also offers communication via multiple channels including e-mail, voice and video communication; (2) The *electronic catalogue* that contains information about all the products and services offered by the virtual shop. Electronic catalogues can have different presentation formats, customised to different types of products, and satisfying the customer preferences for viewing these catalogues; (3) The *surfer that* allows customers to intelligently surf the electronic catalogue, recording their sales preferences in their own *profile; (4) Options for c*ustomers to place buy *orders* online and to *enquire* about the history of their order transactions; (5) *online contact with suppliers and partners*, facilitating inventory control, replenishment, and management; (6) *Online handling of Shipment* orders to assure an adequate stock level of all items and products in the electronic catalogue.

markets, additional features and capabilities are being added to ERP systems, extending their role beyond integrating and automating business processes. These new features include customer relationship management[5] (CRM), enterprise Web portals[6], supply-chain collaboration[7], business intelligence[8], data warehousing, Web-enabling ERP[9], and other added value features [Twe99].

A successful integration between e-commerce and enterprise resource planning (ERP) applications is a crucial factor in effective e-commerce. However, despite the growth and rising demand for e-commerce activity, the corresponding tools and applications are still residing at the front end, and remain somewhat divorced from the back-end enterprise resource planning applications that handle payroll, project planning, business processes,

---

[5] Customer relationship management (previously introduced in chapter 2) is referred to as the ability to capture customers and to satisfy all their needs and requirements with minimum cost and high efficiency. This relies on many tools and technologies that can capture all the relevant information about customers and their needs via multiple channels (e.g. telephone, fax, internet). This information is then stored in databases, and is analysed using data-mining and business intelligence tools to detect profitable customers and to turn the normal call center into an intelligent one.

[6] An ERP portal provides users with a single home screen (web-based) from which they can access all the ERP applications and data sources they need to do their job. The ultimate goal of enterprise web portals is to provide companies with a more efficient channel for delivering self-service applications [Eas99].

[7] In commerce and industry a great deal of effort is made to establish an end-to-end electronic supply chain. This will potentially help in maintaining minimal stock while satisfying customers needs, and in providing low-cost efficient communication with partners. Early attempts at supply-chain collaboration using internet-based trading systems failed due to the difficulty of integration with existing enterprise resource planning applications, the high cost incurred, the lack of security, and the high risk of system failure [Gur99]. Supply-chain collaboration has evolved with more standardised, flexible and efficient electronic interchange systems and technologies.

[8] Business intelligence applications simulate human expertise and reasoning, and operate on massive amounts of data. Within this category we can identify OLAP (Online Analytical Processing), data mining, reporting and analytical applications.

[9] Web-enabling ERP applications means providing a common browser interface to allow employees, customers and partner access to personnel detail, inventory information, or other information in internal systems. Web-enabled ERP is the extension of existing traditional ERP applications to the internet. Web-enabled ERP applications use a thin client architecture rather than a client/server architecture [Eas99]. The main driver behind web-enabling ERP applications is the maintenance of a low cost of ownership in the wide deployment of these applications. Full featured desktop solutions, often called fat clients, are expensive [Hum98]. The total cost of ownership, a common measure of return on investment, brings into a single view all related costs of a technology including capital, administration, technical support, and end-user operations. However, despite the great interest in web-enabling ERP applications, there are still many challenges. These challenges are faced by the current approaches to integration overviewed in the following section.

accounting, and intelligent processing of data. Many attempts have been undertaken to integrate e-commerce and ERP applications. The challenges facing this integration vary in type and scale depending on the technology used at the back-office. Integration with legacy systems, as compared to the integration with modern ERP systems via middleware, poses many challenges.

The complexity of the integration problem presents a challenge to software engineering in respect of system representation and analysis. There is a need for new modelling techniques and principles that can cope with the complexity of the interaction between programmable components and human agents. Viewing the internal ERP system from a web front-end entails many complications that pose fundamental challenges to currently available ERP technologies.

E-commerce applications need to interact with a number of ERP applications (such as sales, order management, payment, fulfillment and customer administration). Integrating a virtual shop model with a suite of ERP applications including purchase, payroll, accounting, and inventory control applications is depicted in the figure below.

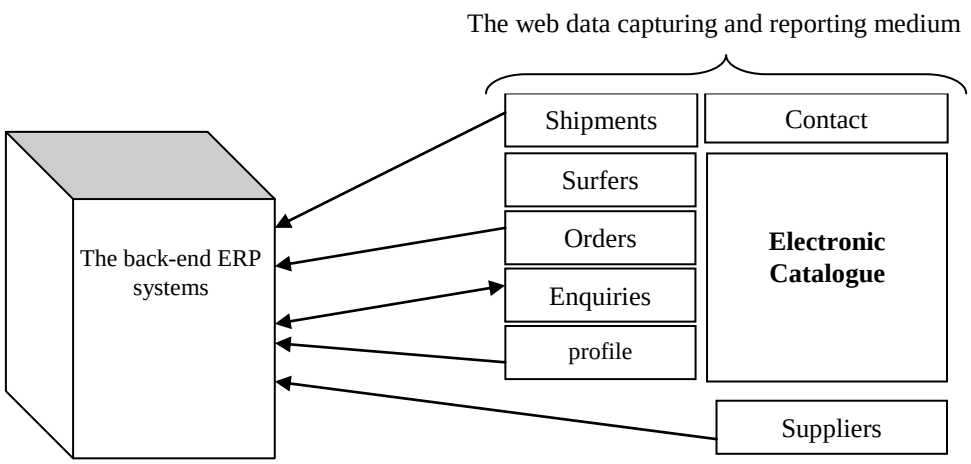


Figure 5.2 Integration of e-commerce (Virtual Shop Model) and ERP applications

Where this integration is not established, the data captured from the web front-end fill-in forms is mediated manually or semi-manually to the ERP purchase application. The response to online customer request for their transaction status or any other information is served via a separate communication medium such as the fax, post or phone, with a considerable delay in the execution process. Moreover, unexpected purchase orders for some products may be delayed if the electronic supply and buy chain is not closed.

The integration of e-commerce and ERP applications provides a straight through processing of customer orders and enquiries and enables stores to hold the necessary level of inventory.

In this context, the integration of e-commerce and ERP applications is viewed as encompassing: web enabling ERP applications, extending and improving the online customer service and the online reporting, and the straight through processing of customer orders and enquiries.

This integration has been approached from many different perspectives. The challenges faced vary in type and scale depending on the technology used at the back-office.

- *The integration with legacy systems*: Connecting web front-end to old legacy systems is very difficult, because these systems make no distinction between the stored data, the structure of this data, and the application. It is hardly possible to modify the data without using the legacy application, which is often undocumented and requires a lot of time and expertise to understand [Han99]. An easy, but unsatisfactory, solution adopted by many firms engaged in the integration of legacy ERP applications with e-commerce front-end applications is to take the information captured online and retype it into the old systems. Another problem facing the integration of e-commerce applications with legacy ERP systems is the need to synchronise the old batch processing of data with the online flow of data captured via the internet browser.
- *The integration with modern systems:* Modern ERP systems consist of two logically distinct elements: a database (which holds the information) and an application (which does the processing). Data stored in databases is then easily manipulated (extracted, filtered, sorted, etc.) using a query language, without the need to understand the application [Han99]. Recent attempts at integration involve linking modern ERP systems with the web front-end via middleware[10].
- *Integration solutions offered by ERP vendors:* Many ERP software vendors are web-enabling their ERP applications, allowing access to these applications from a desktop or a browser interface. However, this approach lacks flexibility and customisation and no one can predict to what extent the problem of shared access to information is properly resolved.

[10] Middle-ware is software that sits between the data sources and applications.

In all the above-mentioned attempts at integration, major common problems are faced. These are mainly due to conflicts arising from shared access to data, poor data quality[11] and rigid architecture and design[12].

5.1.2 Systems Integration Perspectives and Issues

In its broad sense, ***integration*** refers to the coherent merging of two entities, having different behaviours and attributes, to obtain a unified entity that can realise the behaviours of its components and whose attributes are derived from but are not necessarily the same as those of its components. Integration might be necessary to accommodate a new style of relationship between different entities, or it might be optional with the aim of enhancing performance and gaining value-added advantages.

The term integration is used in many different contexts; it can refer to economic integration, horizontal or vertical integration of companies, software integration, system integration, etc.. We concentrate here on the integration of IT systems, focusing mainly on software integration over a distributed hardware configuration.

The first precedents for IT integration are to be found in the databases of the late 1960s and early 70s. At that time, databases typically displaced suites of application programs based on different file record formats and assumptions about physical storage [KS86]. They also reduced the need to commission new programs or re-engineer existing programs to achieve new functionality. The key principles that emerged in this process were *logical data independence* and *physical data independence*. Arguably it was also established, whatever limitations of relational databases have subsequently become apparent, that relational theory and the analysis of functional dependencies in data are an essential aspect of maintaining successful data independence.

[11] These problems arise from the inaccurate recording of information in the internal systems and from much duplicate data [Han99]. Connecting web front-end to ERP applications might violate the data integrity and reduce its reliability if data quality issues are not carefully addressed.

[12] Despite their popularity, ERP applications are criticised for their monolithic structure and inflexibility, the difficulty of combining them with third party bespoke components, the difficulty of component-wise upgrade to new releases, and the high cost of their customisation, maintenance, and training [SSD98].

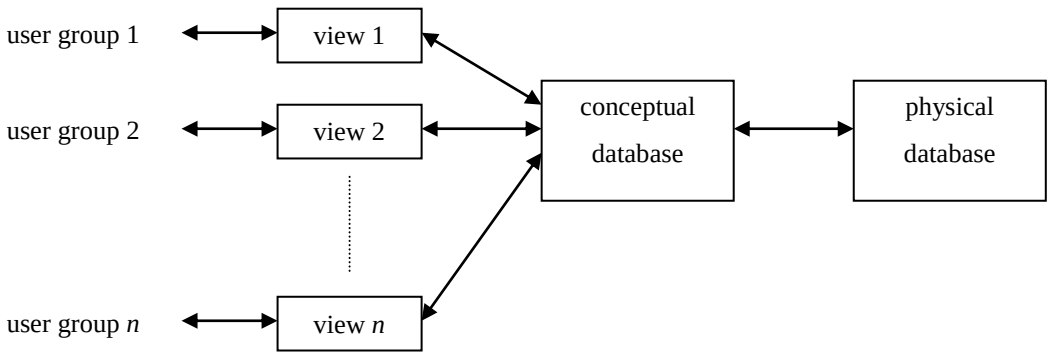


Figure 5.3 Logical and physical data independence

It is now clearer than it was at the time that relational theory supplies a database solution well-suited to a particular kind of business process model. The success of relational databases in areas such as banking depended on the highly routine nature of the transaction processing and the uniformity of the data representation demands. Subsequent developments in computer applications have exposed the need for greater flexibility in database technology than current commercial relational products have been able to deliver. The problems of integration in contemporary applications such as financial systems are further complicated in a variety of ways:

- whereas in the 70s data input was typically manual, and there were no critical real-time issues to be considered, automatic data acquisition via programs or sensors is now common;
- information is now accessed and processed for interpretation in far more complex ways, e.g. through graphical user interfaces, Business Intelligence tools and report generators;
- there is now a demand for large-scale integration of what were formerly quite distinct divisions of business activity, as in the trend to concurrent engineering, datawarehousing and comprehensive business process models.

Where state-of-the-art financial systems are concerned, these problems are compounded by the factors discussed above that promote volatility and instability of the business process model.

Classical approaches to software integration are typically based on combining two paradigms: the use of relational methods to construct integrated data models, and the use of object-oriented methods to describe the operational processes surrounding business data at many levels of abstraction. Both relational and object-oriented modelling methods can be seen as addressing data and agency to some degree.

Functional dependencies in a relational model express the way in which a change to one item of data has to propagate change to other items in a way that is conceptually indivisible if data integrity is to be respected. In the relational theory of database design, they also supply the framework for organizing data to meet the needs of different users. The characteristic mathematical abstractions of relational theory do not capture the distinctions associated with different kinds of agency on the part of the user, however. A relational algebra expression can be used to express how one relational table is derived from others, but this may be used to evaluate a one-off query, to construct a virtual table or view, or (e.g. if associated with a spreadsheet interface) be subject to a process of continuous update in response to independent interactions with its operands. Where the human agent is in full control - as in traditional database applications - distinctions of this nature can be effectively managed without reference to explicit models of agency. Where the database is interfaced directly to computer programs, or to electronic devices with an external interface, the need for an explicit way of expressing and analyzing agency becomes evident. For instance [CW98], two COTS programs that access the same database table may operate quite effectively in isolation, but fail when integrated because one locks the entire table in order to access a single tuple.

Object-oriented models focus upon modelling collections of data together with the fundamental operations that can be applied to them. In this respect, they are well-suited to representing the components of distributed systems, as is appropriate in a typical context of virtual integration. What is lacking is an effective way of dealing with the complications that arise from concurrent interaction between objects. This problem has both *accidental* and *essential* aspects in the sense of F P Brooks (1995). It may be that appropriate re-engineering and the use of frameworks such as CORBA and DCOM can resolve the accidental complexities introduced when processing over several machines in a network, using different programming languages, or running on different platforms. To apply these techniques effectively, it is still necessary to resolve paradigm differences, as when trying to convert a standalone legacy system designed without object-orientation into a modern three-tier architecture. Even supposing that this can lead to a homogeneous collection of objects communicating seamlessly, essential problems remain. It is necessary to account for active

objects and autonomous agents and for the perspective that subject-oriented programming provides [HO93]. There are challenging and well-established problems concerned with the operational semantics of concurrent object-oriented systems. A strong body of evidence from relational database theory also argues against unsupported object-orientation as a solution to data and application integration [COM94].

Issues concerned with agency are amongst the most difficult to represent and the most subtle to analyze and resolve. Operating systems provide the setting in which such issues have generally been encountered hitherto. It is hard to prescribe automatic solutions for the common problems of contention, synchronization and conflict that can arise when many applications are integrated. Experimental activity and insight specific to the particular situation typically rule out full automation. This is acknowledged in the design of GENIO, a software tool for data integration that has been described as a "databroker" [Por99]. GENIO gives the user the means to control the scheduling and propagation of change between data representations either through direct personal intervention or by supplying parameters for automated data conversion agents.

5.1.3 Framing the challenges of software integration

The most important issue in software integration is the plethora of ways in which data is accessed and processed. This can be interpreted as a need for better models of data and agency. The focus is no longer on abstract data alone, but on the state-changing activities that surround that data, to include the protocols and interfaces of all the agents that operate upon it. In this context, an *agent* can refer to a computer program or procedure, a human agent, or an electronic device that mediates between the internal representation and the external world. With this interpretation, agency is manifest at many levels of abstraction, in low-level data exchange, in internal business interactions, and in the external business environment. In general, the problems of integration cannot be resolved without taking account of the multiple views imposed upon data through different types of agency. Only in quite exceptional circumstances, when there is an unusual degree of consistency in the ways that data is addressed and modified by agents, is integration of data representations sufficient. Several key issues have to be addressed for successful IT integration:

- dependency and the indivisible propagation of state change;
- the association of data with operations upon that data;
- the modes of agency by which state-changing activities are mediated and synchronized.

Successful integration gives users and programmers concurrent access to many software applications in a distributed environment (Figure 5.4). It must respect the integrity of data in relation to user views and the external environment. It involves combining interfaces both at the user level and at the machine level, where it provides shared access to raw data stored in system files, taking account of priority and effect. Whilst each of these issues has been addressed individually by current approaches to IT integration based on a relational, object-oriented or data-broker models, there is a need to combine the qualities of all three.

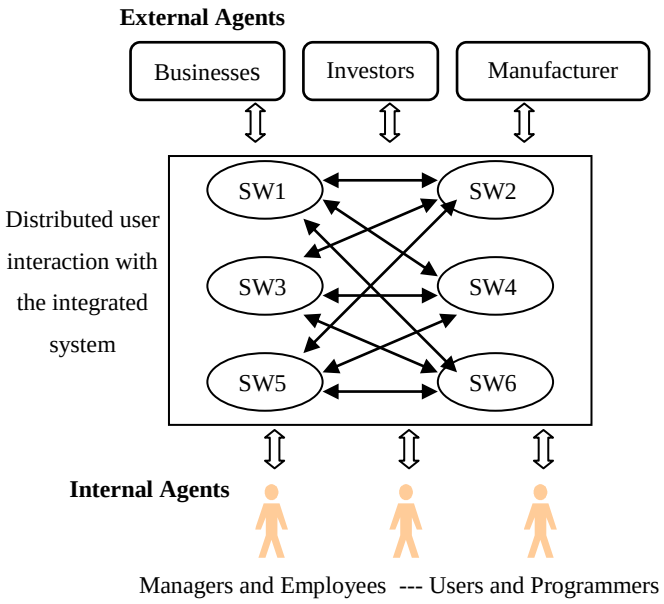


Figure 5.4 An integrated agent oriented system

5.1.4 The Situated Integration Model (SIM)

Software integration in general combines two kinds of activity: integrating existing separate software products and applications, and constructing new multiple-purpose software components. In this exercise in software re-engineering, the principal agenda is:

- interface design and interaction;
- shared access to data;
- synchronization of extraction, transformation, and loading of data;
- creation of a coherent and unified suite of functions.

Current proposals for software integration involve creating a metadata repository comprising profiles of each of the different applications to be integrated. Each profile is compiled from existing documentation and from the results of manual or automated code

analysis. When combined with suitably engineered Common Information Models, the metadata repository supplies the resources from which the integrated system is to be developed.

The creation of a metadata repository is realistic only in certain contexts. It ideally requires the source code, if not the requirements and design documentation, for each component application. The analysis activity is an exercise in program comprehension that can be very challenging if it involves a mix of programming paradigms, or ill-documented legacy code. At best, the repository provides such documentation as is associated with requirements capture and specification in a modern software development method. This provides a static view of the system supported with static workflow models. Even where this information is sufficient to specify the behaviour of individual software components in detail, this does not address the agenda for software integration identified above. Successful integration requires crafting of the corporate behaviour of component applications, to include the specification of interfaces, strategies for shared data access and synchronised data processing, and the design of a coherent functionality.

EM principles for software integration are well-oriented towards the key issues of data dependency discussed in the previous section. Definitive scripts deal explicitly with dependency and indivisible propagation of change. The definitions in such scripts can be grouped in many ways, as their order is not important. By collating the state observables of an EM agent, as specified in LSD, an object-like abstraction is obtained. The permissible operations on such an object in general include redefinition of its state observables through the direct action of other agents. The operational effect of concurrent agency can be empirically established by simulating the execution of agent protocols, for example by using the EDEN interpreter. In this way, two or more modellers can play the roles of agents within the system independently.

Definitive scripts provide a powerful means of data integration that can be used in particular to express the way in which low-level redefinition can entail high-level change. Consider for instance how the interest rate can change when a balance crosses a threshold. Data conversion agents that are empirically tuned to particular patterns of synchronization can serve as databrokers. In EM terms, what has been described in Figure 5.1 as the conversion of data to knowledge is merely one aspect of pervasive mechanisms and processes that mediate between the viewpoints of one agent and other.

In EM, constructing an ISM addresses requirements understanding for software development in a way that circumvents the limitations of normal documentation. An ISM represents

knowledge in an implicit and experiential manner. The modeller can develop, access and explore understanding through interaction with the ISM, and share this insight by presenting the ISM to another modeller. ISMs constructed using EM principles are so general as to encompass traditional engineering or scientific artefacts that are devised to capture empirical insights. Experimentation with legacy components of a software system can be used to develop ISMs in a similar way.

In applying EM to software integration, the key idea is to understand each software application in agent-oriented terms with reference to the particular observables that mediate its interaction with other agents in the system.

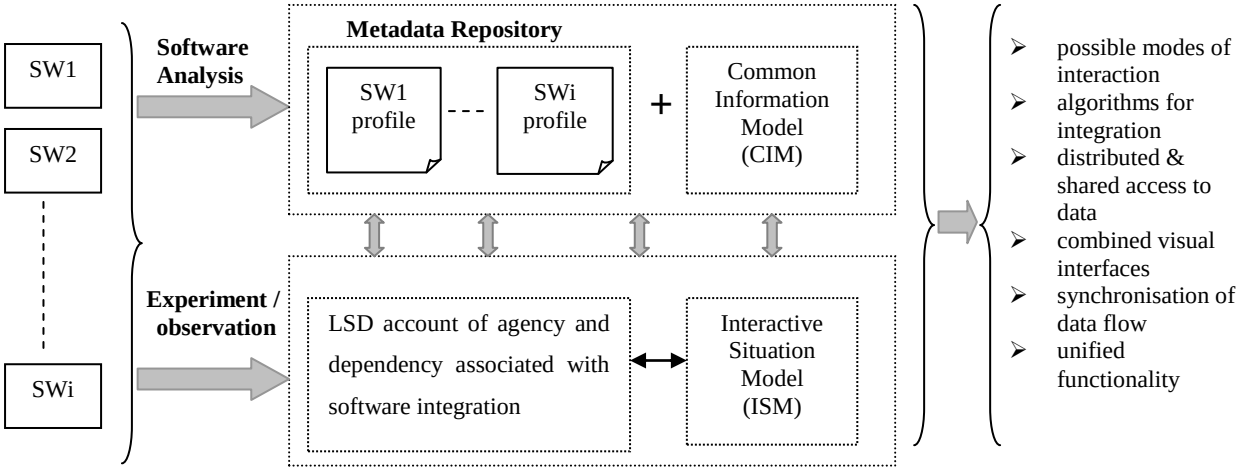


Figure 5.5 The use of EM to complement the conventional approach to software integration

This analysis is not simply concerned with abstract inputs and outputs, but with the way in which interaction is embodied at the interfaces to other applications. This embodiment of inputs and outputs is metaphorically represented by an ISM and its associated LSD account, as developed in parallel. Such ISMs enable an experimental study of the modes of interaction between both existing software applications and those yet to be developed (Figure 5.5). They can also help to address issues of scalability and customization.

The development of an ISM for software integration, to be referred to as Situated Integration Model (SIM), can draw upon ideas and techniques introduced in previous research. These include:

- the construction of ISMs based on animating conventional static artefacts (such as object models and statecharts), both to introduce models to the data repository and to refine and to exploit them [BCSW99];

- collaborative interaction of potential users of the integrated systems (such as the internal and external agents) through networked ISMs in a distributed environment [BS99];
- the intervention of the modeller in the role of an all-powerful agent (e.g. to shape the synchronization of interaction, to resolve conflicts between viewpoints and to compensate for incomplete rules for data brokering) [Sun99].

The EM approach also promises to deliver a model of an integrated system. This model is typically neither static nor comprehensive in character, but comprises a loose association of ISMs constructed from the viewpoints of different agents within the system, each reflecting different observables, dependencies and types of agency. Integration is achieved through a dynamic empirical process of negotiation between these viewpoints. This process in general entails compromise, and may require intelligent intervention by a human agent acting in the role of an arbitrator or broker.

5.2 Virtual Collaboration In The Financial Enterprise

5.2.1 Forms of virtual collaboration

The term *virtual collaboration* refers to collaboration via an electronic medium. The following paragraphs overview different forms of virtual collaboration, and discuss their uses and limitations [BM00].

a) Project/group work collaboration

In current group-ware, documents and document-related processes define the logical context for collaboration [Mar98]. Documents are where most corporate knowledge is captured, hence the importance of document-centric collaboration that is more structured. When teams collaborate on a project, the results of the collaboration are typically captured in documents that need to be maintained and managed. Technologies for collaboration then centre around tools to create, share, and distribute documents. The internet is the most common platform for document-centric collaboration. However, as observed by Ciancarini et al (1999), the web in its current state does not provide support for document-centric applications like group-ware or workflow that require sophisticated agent coordination. In this context, the term *agent* refers

both to entities which can act autonomously and can receive/send messages according to some pre-defined protocol, and to human agents with assigned roles in the group work activity or project.

b) Collaborative learning

Early attempts at online collaborative education were motivated by the desire to explore technical advances in networking and communication rather than by well-defined educational goals. Experience has shown that generic network tools, such as e-mail, computer conferencing, and newsgroups, are weak in supporting collaborative learning [Har99]. This is attributed to several factors. These include: the lack of standardized ways to organize educational material; the overhead work to manage and monitor students' performance; and the lack of models to support learning strategies that involve knowledge building and sharing. Current online educational tools support collaborative learning and course management to a greater extent. These tools are mainly web-based, and include personal workspaces for students, course structuring, grade management, file management, and system management utilities. The Virtual-U web based learning environment (see www.vu.vlei.com) is one example of online learning tools. In addition to online educational tools, research and assessment tools have also been developed to study and analyse the behaviour and teaching/learning processes. Today the main challenge facing the development of web based collaborative learning environments is to support interaction with web pages that is richer than mere front-end access to static information.

c) E- business

The telecommunications revolution and the growth of internet activity have challenged traditional business models by offering direct routes to market, reducing barriers to entry and increasing the efficiency of trade activity [TW99]. E-business is an umbrella term for e-commerce, supply-chain collaboration, online trading, and business to business online communication. Business today is converging on the internet. This creates a great opportunity for organizations to communicate and share data over the web with customers, partners and suppliers. However, the growth and profitability of e-business activity is inhibited by many technical and social problems, including:

- technology integration (the integration of the back office and front office systems)
- the adoption of a common e-business model
- security

- the introduction of new national and international legislation that protects the rights of all business parties in executing cross-border transactions
- the deployment of a low cost efficient solution for true flexible business collaboration
- the high risks of system failure.

5.2.2 Human information behaviour and information horizons

Following D. H. Sonnenwald (1999), the term *human information behaviour* is used to refer to collaboration amongst individuals engaging with information resources in information exploration, seeking, filtering, use and communication. Sonnenwald discusses human information behaviour with reference to three basic concepts: the context, the situation and the social network:

- the *context* is the general setting within which an individual's interactions take place. Academia, family life, citizenship, clubs etc. are examples of contexts. A context is defined by a set of past, present and future situations.
- a *situation* is a particular setting for an interaction within a context. Teaching a course or attending a committee meeting are examples of situations within academia.
- a *social network* is defined by characteristic patterns and resonances of interaction between individuals within a context. In academia, the social network associated with teaching activity might comprise professors, lecturers, teaching assistants, secretarial and technical support staff and students.

In Sonnenwald's view, the goals of a collaboration are the sharing of meaning and the resolution of a lack of knowledge condition. For each individual, collaboration within a given situation and context is bounded by their *information horizon*, as defined by the variety of information resources upon which they can draw. In investigating virtual collaboration, there is an important distinction to be made between information resources that can be accessed electronically, and those that are accessed by other means. In effect, each individual has both an information horizon and a *digital information horizon*.

In Sonnenwald's account of human information behaviour [Son99], each context has its own families of characteristic observables. For instance, in academia, the degree programme, choice of module options and examination marks associated with a student are observables, whilst the weight and height of students are outside the scope of concern. A situation within a context typically includes other pertinent observables that reflect a special focus. For instance, in teaching a course, there is a curriculum, a relevant lecture schedule, and a current point that has been reached in its delivery. Dependencies amongst observables are crucial in shaping the

semantics of a situation. For instance, examination marks attained and the current point in the semester may together determine the possible choices of module options, or the entitlement to transfer to another degree programme.

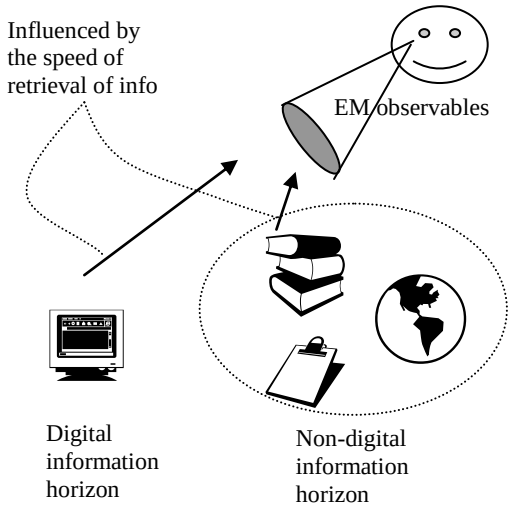


Figure 5.6 Observables and the information horizon

5.2.3 Challenges to virtual collaboration

There are many technical challenges to be met in providing support for virtual collaboration. These include complex and dynamically evolving requirements, as motivated by several key issues [BM00]:

Customisation: Electronic support for collaboration has to take account of the needs of the individual within the social network, situation and context. Identifying and developing algorithms and interfaces to support such behaviour typically requires a high degree of customisation. In particular, electronic support must be well-adapted to the information horizons of the participants.

Integration of the electronic and human activity: Human information behaviour necessarily involves a close interplay between human and automated activity. It is essential that virtual collaboration retains its situated character, so that the information processing activity is appropriately matched to the state of the external world.

Adaptation: The extent to which it is possible and desirable to automate human information processing capabilities is highly dependent upon personal, technological and social factors. The information horizons of participants are typically neither static nor easily

preconceived. They can also be influenced in a deliberate way by the actions of participants. Because of these factors, the requirements for a virtual collaboration are subject to continuous evolution.

A major obstacle to successful virtual collaboration is a fundamental mismatch between the roles that humans and electronic devices play in communication and interaction. This is well-illustrated in current practice by products such as document-centric environments for collaborative work. For the human participant, a document is in general full of significance that eludes formal computational representation. Its meanings are rich, ambiguous and contextually determined. In communication about documents, the human interpreter generally exercises discretion, checking the integrity of interpretations with reference to external observation, or feedback from the person or device with which they are communicating. In contrast, an electronic device records and transmits information according to formal pre-conceived conventions, and - if it monitors external state at all - does so in ways that are highly constrained. Static conventions for representing information limit the extent to which the significance of external experiences can be electronically recorded and conveyed. An electronic device is subject to act without discretion, oblivious to its environment. This can lead to catastrophic failure should singular conditions arise.

The traditional approach to resolving these problems of mismatch is to constrain the interaction between humans and electronic devices to patterns for which a very high level of consistency can be guaranteed. As the above analysis has indicated, this approach is not well-suited to the volatile practical demands of effective virtual collaboration. Its limitations are apparent in all three applications a), b) and c) introduced above. Unless these applications operate in stable environments where consistent patterns of interaction can be identified and exploited, the analysis of content and communication in document-centric collaborative work environments is primarily syntactic; the evaluation of user input in intelligent tutoring systems is stereotyped and semantically superficial; automated decision-making in e-business environments is inadequately guided by the high-level interpretation of actions.

Constraining interaction so as to guarantee reliable and consistent responses from devices affects the quality of human contributions to collaborative activity. Experiential and situational elements play a vital part in human interaction. In a virtual interaction, '`no response`' admits quite different interpretations from '`no response`' in a face-to-face encounter. Such issues motivate the integration of different communications technologies, such as telephones, computers and set-top boxes. To explore this integration effectively, it is not enough to view electronic devices and their interaction in abstract computational terms.

The appropriate emphasis is on electronic components and software applications as mediators of state and experience. Successful integration in these terms entails the assimilation of devices and applications into their environment as instruments. Alternative principles suitable for studying automatic agency from this perspective are not only relevant for developing systems to support human information behaviour. They have an essential part to play in the evaluation of environments for virtual collaboration. They can also be used to assess the intrinsic limitations of existing systems and applications. It would be patently absurd to try to integrate computation with batch cards into a fly-by-wire system, but it is more difficult to assess whether, for example, current web and database technology is appropriate for virtual reality.

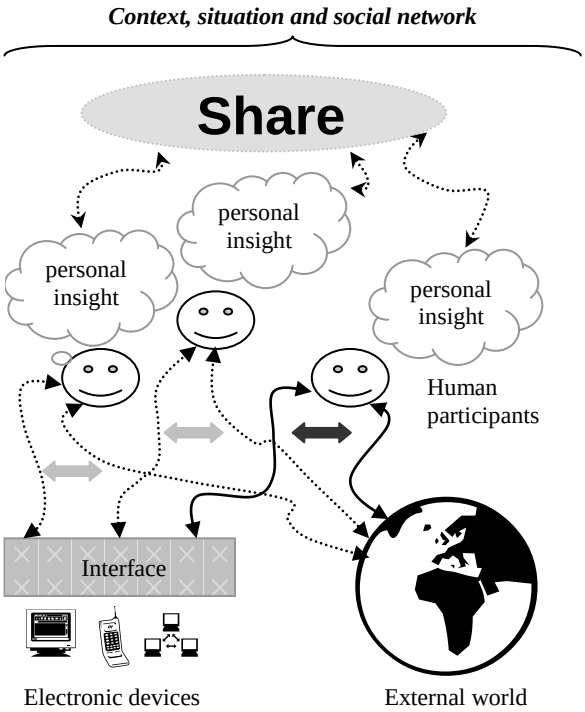


Figure 5.7 Correlating states for successful virtual collaboration

5.2.4 New principles For Virtual Collaboration

Empirical Modelling is centrally concerned with framing and communicating explanations for phenomena. In the context of virtual collaboration, an explanation refers to a convincing way of accounting for perceived state-changes in terms of the interactions of agents. The key questions in this connection are: *What agents are deemed responsible for state-change? What are the cues for state-changing action on the part of agents? What are the direct effects of*

agent action upon the environments of other agents? Contriving such an explanation requires evidence that is typically gathered from observation and experiment. Subjective and pragmatic judgements are involved in interpreting this evidence. It is not in general possible to give a comprehensive account of a phenomenon in terms of agents and their interactions. Patterns of agency and dependency that can be reliably identified as part of an explanation can be framed as an LSD account. The evolving understanding of a phenomenon that eludes even such partial explanation is captured through developing an ISM. This ISM serves a similar purpose to the physical artefacts that an experimental scientist or engineer might construct in order to express their knowledge of a phenomenon. The ISM can be regarded as representing the phenomenon in the informal sense that experience of interaction with the ISM and with its referent are perceived as having characteristics in common. Creating an ISM is of its essence an open-ended activity in which the modeller can venture to embed ever richer perceptions of observables, dependency and agency. Adopting the terminology introduced by Gooding (1990), an ISM serves as a `construal` of the phenomenon to which it refers.

EM principles can be used for constructing ISMs both as an individual and as a corporate activity. Construction of an ISM by an individual has intimate connections with learning activities [BRSW98], and corporate construction with the growth of shared understanding [SB98]. The roles played by ISMs and LSD accounts in EM represent complementary aspects of experimental activity. The LSD account is a way of framing an explanation; the ISM provides an environment in which to explore and evaluate an explanation. EM activity may involve first framing an explanation in LSD, then generating an ISM as a test environment. Alternatively, it may involve constructing an ISM that can be used to explore possible explanations. In general it is appropriate in EM both to use prior knowledge and to seek experimental insight, and – to this end – to frame LSD fragments and incrementally construct ISMs concurrently.

From an EM viewpoint, sharing explanations and understanding is the key to effective virtual collaboration. In relation to group project work, research in [ABCY94] and [BACY94] have examined the potential advantages of using EM both as a way of reaching consensus in design and resolving conflict in creative partnership and – simultaneously – as a playground for individual experiment. In computer supported education, an ISM can be used both to capture personal insights, and as a vehicle for exploring and communicating understanding (cf. the models of heapsort discussed in Beynon et al (1998)). Current EM research in e-business indicates ways in which EM can be used to investigate how human and automatic agents can

co-operate through patterns of workflow and in decision support. The following figure illustrates the relationship between the ISM and the Human Information Behaviour introduced in the previous section. It suggests the construction of Situated Human Information Behaviour Model (SHIBM) based on an ISM that captures: (a) the description of observables representing the context of collaboration; (b) the state of observables representing the situation in collaboration; and (c) the role of agency (introduced as dependencies in the ISM script) in changing the state of observables and establishing a semantic relationship between the digital and non-digital information horizon of individuals. The Human Information Behaviour is implicitly expressed in the interaction with the ISM, and explicitly described in an LSD account.

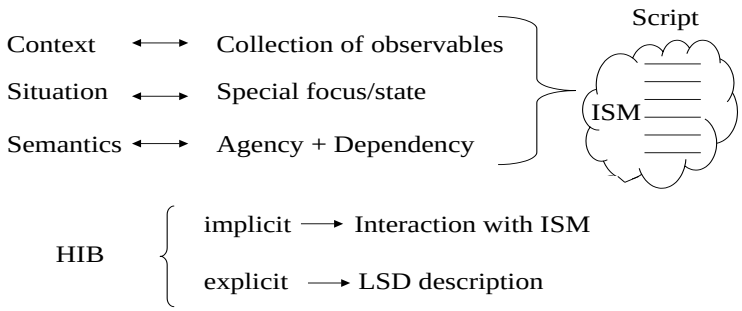


Figure 5.8 A Situated Human Information Behaviour Model (SHIB)

5.2.5 Virtual Collaboration in Online Trading

Different models were presented in [BM00] to illustrate the application of EM principles to various forms of virtual collaboration. For collaborative learning a VEL model (previously introduced in chapter 4) was considered. For collaboration on group projects, examination assessment in the academic context is considered. For business to business collaboration, online trading was considered. This section reviews the latter model of collaboration in the context of a retail trade in NYSE[13].

In modelling an online trading environment, collaboration can be viewed as a workflow[14] of interdependent tasks undertaken by human and electronic agents. A situated human

[13] The story of a retail trade in NYSE is taken from [Har98], and was previously considered in chapter 3 to illustrate the construction of an ISM and the use of LSD notation in EM.

[14] Research in the area of workflow management systems in a business context attributes the difficulty of virtual co-operation between organizations to the lack of standard ways of representing an application's structure and sending and receiving work items [SWH99]. Although the extended markup

information behaviour model of online trading consists of an ISM that supports the exploration of the electronically mediated interaction and communication of human agents involved in the online trading activity. The information horizon of various agents involved in the trading activity is implicitly represented in the interaction with the ISM and explicitly described in an LSD account.

The ISM described in chapter 3 is very simple in nature: it does not take the actual character of the transactions and interactions into account, but merely registers the pattern of the workflow. A more sophisticated ISM would aim at exploring possible scenarios that can arise in the retail trade process (RTP). These scenarios are much more subtle than the workflow alone indicates: in real practices transactions may be disrupted by communication failure, by human error, or by dishonest dealing. The RTP may take place in a setting where other kinds of observation pertain. There will be a stage at which the investor is legally commited to complete, for instance. These broad issues regarding the RTP will have to be reflected in devising a useful ISM.

The potential subtlety of the RTP is mirrored in the possible interpretations that can be given to the LSD account of the broker agent, presented in chapter 3, and the elaborations that these motivate. The LSD account refers to `info_requested` as both a **state** and an **oracle**. This highlights a potential ambiguity concerning a particular information request. As an **oracle**, `info_requested` refers to an observable that is associated with an investor. This can be interpreted as saying that the broker is - or at any rate can be - aware that an investor is requesting information. For the purpose of giving a routine account of the workflow, how such a request is mediated to the broker is irrelevant, and the possibility that the broker may be too preoccupied to note the request is discounted. As a **state** for the broker, `info_requested` refers to an observable whose status is private to the broker. Recording `info_requested` as a **state** potentially admits discrepancies between what the broker believes or recalls and what the investor has declared. The consequences of such discrepancies are implicit in the interpretation of the broker's protocol. The precondition for

---

language (XML) and other similar standards for data exchange, such as Open Financial Exchange (OFX) and Open Trading Protocol (OTP), are significant advances in this area, full co-operation between organizations is still a long way off. Present day workflow systems are not scalable, as their structure tends to be monolithic and they offer little support for building fault-tolerant applications [SWH99]. Software development in the area of workflow management systems is directed towards developing concepts, methodologies, techniques, and tools to support workflow-process management [SAA99]. The main challenge facing the networked economy is to design workflow processes that cross organizational boundaries. This is especially difficult when these boundaries are fluid and subject to continuous change.

action on the part of the broker can be read as: the broker *believes* that a particular stage in the RTP has been reached and that information has been requested.

Similar considerations apply to the **derivate** that determines the stage reached in the RTP. The definition

```
stage_in_retail_trade = F(info_requested, … )
```

is used to indicate that the current stage in the RTP can be construed as functionally dependent on the status of transactions. In a naïve account of the RTP, this can be seen as reflecting the fact that, once the investor has requested information from the broker, a new stage of the RTP has been entered. Introducing such dependencies in the ISM for the RTP gives the assignment of a new value to the observable `info_requested` the quality of a redefinition - an action that potentially has indivisible effects on the state of other observables. Such a mechanism could also be used to take account of whether an action had some legal consequences, such as might express a commitment or obligation. From this perspective, it might also be appropriate to deem `stage_in_retail_trade` as also dependent upon the precise contents of a transaction: if shares were paid for using counterfeit money for instance.

One motivation for embellishing the LSD account and the ISM for the RTP is that many different communication technologies and information strategies can be used in the RTP. As an observable, an information request placed by telephone has quite different characteristics from a web request. What observables a broker uses to determine the current stage of the RTP may be hard to ascertain. A precise procedural account of how a broker processes a request from an investor might not resemble a redefinition, and could quite easily involve creating and then resolving inconsistent states.

By implication, this is not a simple redefinition, but a sequence of related assignments. Conceptually, it is much harder to guarantee the integrity of the state which it creates. It might on the other hand be necessary to formulate the broker's role in this much detail to capture the true situation more faithfully. For instance, if updating the current RTP status involves some explicit book-keeping on the part of the broker, it is possible for this action to be accidentally omitted.

The above discussion indicates the kind of analysis that accompanies the development of an ISM for the RTP. The precise scope and nature of the ISM is open: it could be oriented towards a high-level account of workflow, or to a specific framework for implementation.

The modelling process may be helpful in addressing the integration of human and automatic activities, and could be directed specifically towards related goals. It might also indicate that, in some situations, effective integration is infeasible with current technologies and paradigms, where computation is too far abstracted, and the potential agency of automated components is too limited.

The ISM assists in exploring the ***information horizon*** of human agents participating in the trading process. The information horizon for an investor (cf. Figure 5.9) includes many of the most significant of observables captured in the ISM and described in the corresponding LSD account.

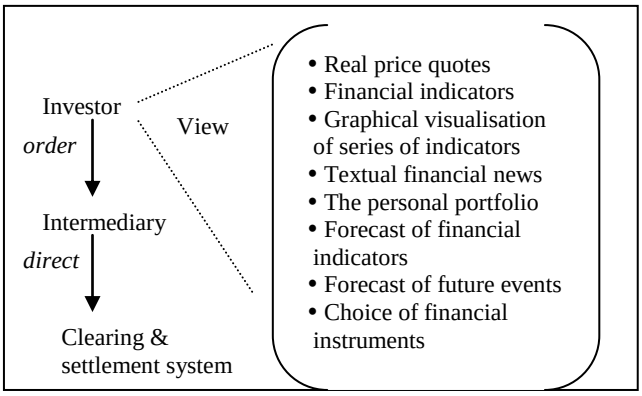


Figure 5.9 A typical information horizon of an investor

Many topical issues motivate the exploration of the information horizon of an investor. Today's investor is looking beyond receiving delayed financial indicators. Trading in a sufficiently liquid and cost efficient market is becoming a major concern for investors [Lan99], and this motivates a better understanding of the trading environment and the layers of intermediation. The support of a large range of instruments, the quality and timeliness of information feed, the functionality of the front-end, and the scalability and performance of the system are important factors in designing digital information resources for an investor. Technology is opening up new avenues for investors to cut out the layers of intermediation and talk to one another directly. This places a question mark over what value can be added to the trading process by the stock exchanges and their constituent brokerages [Lan99]. Current online trading networks provide a huge amount of static information for the investor to interpret and analyse. Online trading web sites have been created by brokerage firms with the aim of extending the digital information horizon of an investor. To this end, these web sites

are currently delivering free access to delayed prices, portfolio management services, and graphic visualisation of financial indicators.

5.3 Summary and Future Outlook

This chapter has investigated the potential application of Empirical Modelling technology in the financial enterprise. EM technology can play an important role in the requirements engineering of software system development for the financial enterprise. Integration and virtual collaboration are applications where EM technology can potentially offer a great contribution in directing a special focus to the central role of human agents involved in these applications.

The need and challenges to software integration in the financial enterprise were motivated with the example of integration of e-commerce and enterprise resource planning application. A Situated Integration Model (SIM) is proposed as an integration model that takes into account the social and technical aspects of the software integration activity.

Collaboration is a situated activity that aims at sharing explanation and understanding amongst individuals. Empirical Modelling technology proposes a computer-based support that takes into consideration the human information behaviour and the information horizon of participants in the collaboration activity. Such computer-based support draws on the construction of an ISM and an LSD description that respectively represent the human information behaviour implicitly and explicitly. The complexity of the collaboration activity and the importance of understanding the roles that humans and electronic devices play in communication and interaction is motivated through the discussion of an ISM for a financial retail trade story. Such an ISM would assist in exploring various scenarios and singular situations that might arise in the course of communication and interaction of human agents with programmable components and devices.

The prospect of Empirical Modelling in the financial enterprise is in delivering computer-based support to activities that are human centred and in which singular conditions and changes are likely to arise. With this distinctive quality, Empirical Modelling technology is potentially well positioned to support the development of corporate intelligent networks that integrate the human and technical activity with greater coherence and adaptability to change. Porting EM principles to web technology is another aim that should be pursued in EM research.

Chapter 6

Emprical Modelling of The Financial Market

6.0 Overview

This chapter proposes an Open Financial Market Model (OFMM) for exploring and understanding the shift from the old to the new trading model. The OFMM aims at establishing a closer integration between the software system development activity and financial trading related activities. The latter activities include: understanding of the requirement of the new trading model, decision support, and modelling of the financial trading process.

Section 6.1 motivates the OFMM with reference to the challenges facing computer-based support for financial trading activities associated with the shift from the old to the new trading model. Section 6.2 proposes basic principles for developing the OFMM. Section 6.3 briefly reviews existing models drawn from academic and practitioner circles and takes one of them as the basis for a case study that can provide a proof-of-concept in respect of the technical implementation and uses of the OFMM. Section 6.4 proposes an EM – VR merge as a contribution towards a broad foundation of computing based on EM technology for addressing the technical and strategic demands of the agenda for computing in finance. Section 6.5 concludes with a summary of the key characteristics of the OFMM and the future prospect of EM technology in providing a closer integration of the software system development activity with the financial trading activity.

6.1 Challenges Facing Computer-based Support For the Financial Market

Trends in trading practice associated with the advent of virtual environments motivate a reappraisal of activities in the financial market domain. Relevant issues include: i. understanding the complex trading environment and the behaviour of its market participants, ii. modelling and re-engineering the financial trading process, and iii. providing decision support for market participants. These issues require deep domain knowledge and the choice of an appropriate computational paradigm.

i. Understanding the complex trading environment and the behaviour of market participants:

The financial market is a complex system involving many different participants (investors, brokers, dealers, market makers, floor specialists, etc.) and a large number of instruments meeting various financial needs and requirements. This makes software system development for the financial market a subtle and dynamic activity. Understanding the trend of various trading signals is difficult and requires profound knowledge of financial theories and practices. This is exemplified in three key activities:

(a) Assessing *the impact of the trading behaviour of different market participants and of the introduction of new trading systems on market prices and the role of different market participants:* The trading behaviour and the role of market participants shapes, and is being shaped, by the financial market microstructure (the execution system and the market information system). The shift from an old to a new trading model implies a new market microstructure and necessitates advanced approaches to understand and study the behaviour of market participants and its impact on the market.

(b) Identifying the constituents of transaction cost: Different estimations of the trading cost are found in the finance literature. The cost of trading a security is given as composed of two parts: an implicit[1] and an explicit[2] part. Haynes (2000) subdivided

[1] The bid-ask spread is the implicit cost of trading. The dealer, who acts as a broker, may charge no commission but collect the fee entirely in the form of the bid-ask spread. Another implicit cost of trading that some observers identify is the price concession an investor may be forced to make for trading in any quantity that exceeds the quantity the dealer is willing to trade at the posted bid or asked price.

trading cost into a visible and hidden part (cf. Appendix 2.3). The visible part includes taxes, commission, and spread, while the hidden part of the trading cost includes market impact and delay. It is difficult to find an exact formulation of the trading cost as many factors that cannot be exactly quantified or assessed contribute to it.

(c) Estimating the true price of securities: Like trading cost, the true price of a security is difficult to quantify in exact metrics. Theories in finance attempt to describe price movements in the market. They include the random walk and efficient market hypothesis, theories related to the determination of the intrinsic value of a security, and prediction and financial analysis theories and practices. A brief overview of these theories is given below:

- Prices in the markets follow a random walk[3] [BKM96]. This does not imply that trades are irrational. Randomly evolving stock prices are the necessary consequences of intelligent investors competing to discover relevant information on which to buy or sell stocks before the rest of the market becomes aware of that information. Randomness in price changes is different from irrationality in price changes. If prices are determined rationally, then only new information will cause them to change. Therefore, a random walk is the natural result of prices that always reflect all current knowledge. The notion that stocks already reflect all available information is referred to as the *efficient market hypothesis*. The general idea behind the three forms of efficient market hypothesis EMH[4] is that, except for long-run trends, future stock prices are difficult if not impossible to predict.
- Many theoretical foundations lie behind the determination of the intrinsic "true" value of the security. Firm foundation theorists view the worth of any share as the present value of all dollar benefits the investor expects to receive from it. In an attempt to answer two questions: what determines the real or intrinsic value of a

[2] The explicit trading cost is the commission paid to the broker. Two types of brokers are identified: full-service (providing full service including financial analysis) or discount brokers (provide no services other than buying and selling securities, holding securities, offering margin loans and facilitating short sales).

[3] A random walk is one in which future steps or directions cannot be determined on the basis of past actions.

[4] The efficient market hypothesis (EMH) introduced by Fama (1970 and 1991) is one of the central ideas in modern finance. There are different versions of the market efficiency hypothesis according to the information set that is assumed to be contained in market prices. In weak form efficiency, current market prices reflect all information on past prices. In semi-strong form efficiency current market prices reflect all publicly available information. Whereas, in strong form efficiency current market prices reflect both public and private information. Malkiel (1996) stated that a capital market is efficient if it fully and correctly reflects all relevant information in determining security prices.

share? and what are the so-called fundamentals that security analysts look at in estimating a security's firm foundation of value?, Malkiel (1999) identifies four determinants affecting share value: the expected growth rate, the expected dividend payout, the degree of risk, and the level of market interest rate. However, these four determinants cannot exactly reflect the true value of the security because expectations about the future cannot be proven in the present and precise figures cannot be calculated from undetermined data.

- Harris (1998) demonstrates algebraically[5] that informed traders make prices informative. The buying of informed traders tends to push prices up and their selling tends to push prices down.
- Ridley (1993), regards market trading as relying on intuitive and complex reasoning on the part of the human trader. The trader interprets and deciphers many factors surrounding the market of interest. The factors can be wide ranging and can vary over time. This kind of changing structural relationship implies that the form of decision process required by market trading is not open to precise calculation and therefore not open to mechanisation.
- Investors conduct two types of analysis to predict future prices: technical and fundamental. Technical analysis studies trends in historical charts to predict the future, and fundamental analysis attempts to estimate a true value of a security and buy and sell according to the difference between the market price and estimated true value of the security. The weak and semi-strong forms of the EMF raise doubts about the effectiveness of the technical and the fundamental analysis. However, investors still conduct both types of analysis to predict the market.

ii. Modelling and re-engineering the financial trading process:

A financial market model would aim at supporting the design of a better market by exploring current and potential market microstructures. Many factors are reshaping the financial market microstructure. These include globalization, integration, trade liberalization, monetary union,

[5] The algebraic illustration of how markets aggregate information considers n traders each producing a different forecast of the true value of the security. This forecast includes an estimation error factor. Assuming that the trader's desired position in a security is proportional to the difference between his/her forecast and the market price and that the security is in zero net supply, then the market price is computed by equating the sum of all desired positions to the net supply of zero. The market price is then equal to the average individual forecasts and the true price of the security is equal to the market price plus the average individual traders' forecast errors. As the number of informed traders get larger the market price will approach the true value of the security.

deregulation, and financial crises[6]. These factors are putting great pressure on financial markets to re-engineer their business processes and maintain a competitive position in a global market. Re-engineering the trading system demands a closer integration of the *trading process model* and the *financial analysis* adapted to the corresponding process model (cf. the figure below).

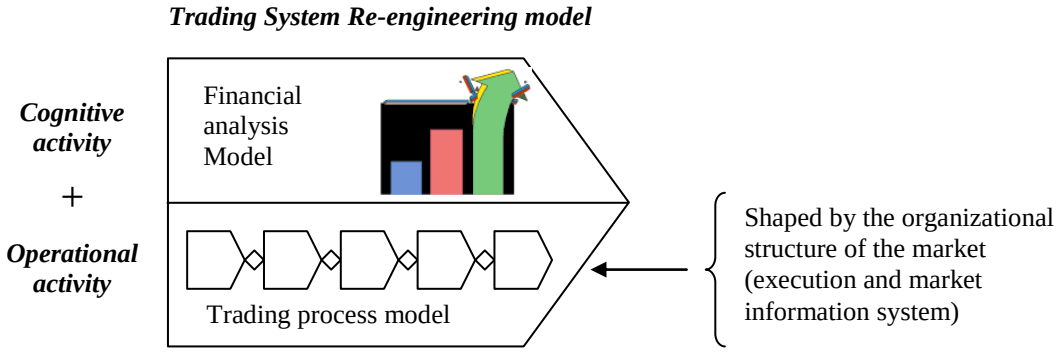


Figure 6.1 Integrating the trading process and the financial analysis models in the re-engineering model

Konana et al (1999) depicts the trading process in a transaction cost/revenue model tracing the workflow of an order directed from an investor to an electronic broker. The broker can then channel the order to a market maker or to a stock exchange. The aim of the model is to reveal the impact of the trading process on the implicit and explicit part of the trading cost. This provides a high level model of the trading process that bypasses the organizational structure of the market and its impact on the order flow and the transaction cost.

Harris (1998) introduced nine trading stories showing the different trading process models adopted in trading a particular type of financial instrument in a particular type of market. These stories include: a retail trade in NYSE, a retail trade in NASDAQ stock, an institutional trade in a NYSE stock, an institutional trade in a NASDAQ stock, a very large block stock trade, a cask commodity trade, an option market trade, a bond market trade, and a foreign exchange trade. We can infer from these trading stories that the price at which a transaction is executed and the behaviour of market participants are shaped by the trading process model and the type of instruments traded.

[6] *The crash of the New York stock exchange in the 1980's* -The largest stock-market drop in Wall Street history occurred on "Black Monday" - October 19, 1987 - when the Dow Jones Industrial Average plunged 508.32 points, losing 22.6% of its total value. That fall far surpassed the one-day loss that began the great stock market crash of 1929 and foreshadowed the Great Depression; *The Asian crisis* - In June-July 1997 the currencies of Asia started to fall. Without exception all Asian currencies have fallen [Tho97]. The impact of the Asian crisis was felt worldwide.

The trading process is highly influenced by the organizational structure of the market and the type of instruments traded. The latter two factors affect the price determination mechanism in the market and consequently the investor's analysis and forecast of the market. Describing a trading process can be approached in two ways: using a literal description or a business process model. A trading system re-engineering model is better targeted when integrating the trading process with the financial analysis conducted in a particular market.

iii. Providing decision support for market participants

In the face of highly competitive and changing financial market conditions, providing decision support to different market participants becomes urgent. It is difficult to devise a formal methodology that informs the decision making of a market participant. Understanding the complex trading environment (assessment of the impact of the behaviour of market participant on the financial market, the identification of the constituents of the transaction trading cost, and the estimation of the true price of securities) challenges formal methodologies in leading the decision making activity. Greater adaptability and flexibility is needed to support the decision making activity and the gradual knowledge construction.

6.2 The OFMM Concept

6.2.1 Motivations and aims of the OFMM

The previous section highlights various challenges that face computer-based technology in supporting the shift from the old to the new trading model. These challenges were revealed by a closer view of the various activities undertaken in the financial market domain. In this section, we explore the possibility of meeting these challenges by developing a novel kind of computer-based financial market model, to be called an open financial market model (OFMM), that promises greater support for the shift from the old to the new trading model.

Openness in the model is proposed at a conceptual and practical level. At a conceptual level openness connotes incompleteness of the model; its use for knowledge construction rather than representation; its ability to reflect growing experiential knowledge; its lack of circumscription; and the absence of a particular methodology to develop the model. Practically, the term 'open', used as an attribute for a model, refers to a high degree of

flexibility in revising the state of the model through adding, amending, and discarding information.

In providing support for the shift from the old to the new trading model the emphasis is on modelling rather than programming. The term *model* differs from the term *program* and the term *software system*. Whereas a program has a specific behaviour usually described in terms of input-process-output, a model (in the sense of the term adopted in this thesis) aims at representing an evolving state. A system is an end product that can be derived from the model through an appropriate circumscription of the model once a desired functionality can be relied upon. It is assumed that the boundary of the eventual system is not pre-conceived but rather grows with the developing understanding of the modeller – in respect of both understanding of the domain and of the requirements for the system.

An OFMM aims at establishing a closer integration between the financial trading related activities and the computational activity, in particular the software system development activity. The model must be essentially open for two reasons: the growing complexity of the financial markets; and the limitations/incompleteness of financial theories to fully explain financial markets phenomena. The use of the model is not restricted to a hierarchy of users. It is open for use by:

- Designers - developers -users participating in the software system development for the financial market. The designer is the financial expert dictating the features and functionality of the financial software system. The developer is the technical expert in charge of constructing the software system meeting the designer's requirements. The user refers to the end-user of the financial software system designed by the financial expert and produced by the developer. This end-user might have some financial and computer knowledge however this is not essential. He / she uses the financial software system to fulfill an operational need or gain knowledge and understanding of the financial market.
- Market participants (brokers, investors, market makers, floor specialists) seeking decision support
- Academics cultivating experiential knowledge of the financial market

An OFMM refers to a physical artefact representing a personal and subjective construal of the financial market. The physical artefact is mainly computer-based, it is constructed experientially as opposed to formally, and its use is tailored to the modeller's insight and

perception. It can be used for establishing a closer integration between the software system development activity and various activities undertaken in real world contexts such as learning, decision support, business process modelling. The model can be shared collaboratively to take account of both private/subjective and public/objective perspectives, to be more realistic, and to better serve group social activity.

An OFMM assists *requirements engineering*[7] in software system development for the financial market. It promises a finished product at some point in time. However, the conditions for the delivery of this product are difficult to preconceive in advance. This assumes that the requirements engineering is not only limited to an early stage of the software system development activity but spans the whole period of development.

It is not realistic to expect to construct a computer-based model of this nature with current technologies and limited experience of the highly changing financial market conditions. We focus instead on proposing basic principles for developing such a model. Three aims motivate the development of an OFMM:

- cultivating the understanding of financial markets phenomena and seeking an appropriate medium to convey this understanding and to communicate it across a social network;
- developing computer-based support that establishes a closer integration between the financial trading related activities and the software system development activity;
- better understanding and support for the shift from the old to the new trading model by developing a computer-based means to explore the new trading model and its impact on the financial market microstructure experientially.

6.2.2 Basic principles

The motivations and objectives of an OFMM introduced above highlight important considerations in choosing an appropriate computational framework for an OFMM. These include:

[7] Requirements engineering refers to the early phases of the traditional software system development cycle. As overviewed in earlier chapters, this development cycle consists of five stages (requirement analysis, specification, design, implementation, and verification). Software system development focuses on the process initiated in capturing the statement of the problem of a client and terminating with the delivery of the software product that will potentially solve the problem. Different computer aided software engineering CASE tools can be used at different stages of the software system development cycle.

- *The experiential dimension of the model building activity*: trends in trading (new trading styles, mixed virtual/real trading contexts, etc) oblige a richer account that conveys experiential knowledge which is difficult to gain through abstraction. Financial concepts admit rich and diverse construals. Such construals should be reflected in any financial modelling framework.
- The agency, observable agenda: financial market modelling requires a far richer quality of observation and agency description. This is of importance and particular relevance to the software system development for the financial market.
- Openness and flexibility: change in the financial market is rapid. This obliges the adoption of approaches to modelling admitting openness and adaptability that do not require wholesale model reconstruction.

Chapter 3 reviewed the distinctive qualities of model building in EM: a) the focus on state as experienced; b) the maintenance of a semantic relationship between an application domain and a computer-based artefact; c) the use of an artefact for knowledge construction; d) the support for collaborative relationships in distributed modelling. These qualities have broad implications for the software system development activity and its computer-based support in terms of openness, situatedness, and the absence of rigid methodologies. Modelling is central, and situated user-developer-designer collaboration is essential for cultivating the understanding of requirements. This points to the conclusion that EM provides a suitable basis for the development of an OFMM. This case is supported below through the consideration of a series of claims. Each claim is addressed by reviewing specific characteristics of the finance domain and by establishing a comparison with traditional computational paradigms when necessary.

Claim 1: ***Principles for experiential knowledge construction in EM can be adopted in developing an OFMM.***
Claim 2: ***Establishing a semantic relationship between an OFMM and the real financial market domain, following an EM approach, can potentially cope with major issues surrounding the new financial trading model. These include the estimation of transaction cost, true price determination, price movement, and the impact of the behaviour of different market participants on the trading activity.***

Embodied intelligence is an important concern in developing an OFMM as it determines the uses, scope, and evolution of the model. Knowledge is needed to perform intelligent (expert) tasks. Encoding knowledge in a model assumes completeness of this knowledge and is used for selecting an optimum solution for a problem. The question to be addressed is whether an OFMM should provide a fixed repository of encoded knowledge, a growing repository of encoded knowledge, or it should support knowledge construction in an experiential way.

EM technology aims at using the computer as an instrument that supports mental modelling. EM technology acknowledges the limitations of human thinking in solving problems in the real world domain. It attempts to support human thinking and decision making, where possible, without imposing a rigid framework shaping this thinking. EM technology takes account of the situatedness of the human thinking and common sense reasoning. In solving problems in the real world domain, EM technology considers the whole problem from a personal, subjective or shared objective view. A divide-and-conquer[8] strategy can be adopted, but the subdivision of problems and tasks is subjective and is made in response to personal or shared insight. EM technology does not assume complete knowledge of the real world problem domain; it attempts to construct this knowledge in an experiential way by interacting with the real world and a physical, typically computer-based, artefact. The key characteristic of this artefact is that its state is linked by a semantic relationship to the state in the real world problem domain and gradually evolves with the state in the real world. A state in EM is defined in terms of observables, dependencies, and agencies. EM supports the construction of knowledge about this domain.

AI technology, based on the physical symbol system[9] hypothesis, aims at using the computer as if it were a human brain. It ignores the limitations of human intelligence, assumes that human knowledge and reasoning are complete and correct, and proceeds to the next step of encoding human knowledge (referred to as knowledge engineering) about a problem domain for future use in exploring a solution space. In considering a complex real world problem domain, AI technology subdivides the activity of solving the problem into tasks and classifies these tasks as ordinary, formal, and expert tasks. From a technical point of view, AI technology adopts methodologies for knowledge management and solution search. The four

[8] A divide-and conquer strategy involves sub-dividing complex problems into smaller sub-problems and devising sub-tasks to solve these sub-problems.

[9] A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure) [NS76].

step process[10] in building an AI system assumes full knowledge about the system before building it and using it reliably in different situations.

Key issues raised in the application of AI to finance are knowledge representation, goals, and methodologies to solve problems in the investment domain. Abstraction, structuring, and model selection [TA83], the size of the problem space [NS72], its character, and the solution methodology in AI, all place economic limits on the scope of the problem that can be solved.

In knowledge-based systems, the solution space is defined by the way in which knowledge is represented. Trippi (et al, 1996) defines *knowledge representation* as the formalism for the systematic computer storage of facts and rules about a subject or speciality. Encoding knowledge in the knowledge base is referred to as *knowledge engineering*. Expert Systems are applied to routine financial decision making operations. The decision making process involves: identification of goals and constraints, generation of a set of feasible investments, formulation of alternative strategies, selection of an appropriate strategy, implementation of the selected strategy, and the explanation of results.

In developing an OFMM, a traditional AI approach would be directed towards building a knowledge base about such issues as dealer and investor expertise and true price generation. Knowledge about a dealer's strategic actions in response to trading flow can be encoded in an expert rule database that is consulted to determine the most appropriate action in response to a trade pattern. Such an approach is similar to the one used by Andrew Martin in developing a management information system (MIS) game [Mar00, MCB99]. Rules for the dealer's strategic actions in response to trade flow can be encoded in the knowledge base. Examples of rules are[11]:

```
IF ((estimated true price > ask) OR (informed trader rush to buy))
     THEN  raise ask
IF ((estimated true price < bid) OR (informed trader rush to sell))
     THEN  raise bid
IF ((spread == wide) OR (few uninformed traders are trading))
     THEN narrow the spread
IF ((inventory < -10,000 ) OR (inventory > 10,000))
     THEN adjust quotes to attract buy and sell orders appropriately
```

[10] (1) defining a problem in terms of input (initial situation) and output (acceptable solution); (2) analyzing the problem; (3) encoding the knowledge necessary to solve the problem; and (4) choosing the best problem solving technique to apply to the problem

[11] Rules encoded in a knowledge base are different from protocols in an LSD description. The key difference is that an LSD description reflects our initial understanding of the real world domain and is subject to amendment with our exploration of the real world domain and our interaction with the computer-based artefact (EM model).

The limitations of the knowledge based approach for developing an OFMM is revealed when attempting to encode knowledge about the true price determination and the rules for dealer's strategic action in response to his/her estimated true value of the security. Encoding rules for the determination of the true price of a security is a challenging task since the true price formation can be construed in different ways and in a situated manner.

Many researchers subscribe to the view that human knowledge is continuously growing in such a way that it cannot be circumscribed in a representation medium [Slo90, Cla97, Suc87]. Knowledge acquisition and construction is a continuous on-going process and cannot be limited in space and time. This undermines systematic approaches to knowledge encoding which assume a complete knowledge of the domain prior to the development of the AI system. Knowledge constructed over time requires extension or alteration of existing formal representations of knowledge. Abstract physical symbol systems used to represent reality might not always be adequate and may need refinement over time.

From an EM perspective, knowledge construction and knowledge representation are two complementary activities and both should be taken into account [Sun99]. Practically, definitive scripts are used to capture initial knowledge of the real domain, and re-definitions are used to introduce newly acquired knowledge. In constructing an OFMM, EM principles can be useful in construing a situation in the financial market context, and in capturing the state of this situation in a definitive script that can be used to realize and explore different possible construals.

> ***Claim 3:*** ***The adoption of an EM approach to situated collaborative distributed modelling in an OFMM supports semi-automated group decision making where human input and insight is central. This gives participants in the financial market greater potential in making decisions based on their constructed knowledge about key drivers in the financial markets, such as the true price of the security, traders' behaviour, transaction cost, etc..***

As discussed in earlier chapters, solving problems in the real world domain is a situated activity rather than a formal activity, especially on the first occurrence of the problems. Suchman (1987) argues that most plans (algorithms, strict laws, formal methods) are used by human agents as a resource rather than a source of control in every day life. This argument is

also supported by the fact that a solution to a real world problem is context dependent, and human centred. However, a situated activity is error prone because it is human dependent, as compared to a formalised process derived from an engineering discipline.

In the context of the financial market, examples of situated activities are the dealer actions to adjust his/her quotes, and investors' actions to buy and sell shares. This situated activity is very human centric, because it relies heavily on the dealer's perception of the market and of trading flow.

Providing decision support to market participants (dealers, investors, market makers, etc.) in an OFMM raises the question of whether a fully automated or a semi-automated decision support system is adequate. A fully automated decision support system provides a solution to a well defined problem. The market participant does not intervene in the solution finding, apart from setting parameters in the input describing the problem and in framing the desired output. The resulting decision is either taken by the human participants or discounted in favour of a personal common-sense judgement. The semi-automated decision support allows the market participant to intervene in the process of identifying the problem and the solution space, and the exploration of a possible final solution. A semi-automated solution might not be the optimal solution as it is guided by human insight that might be erroneous and/or more limited than can be gained with automatic support. However, a semi-automated solution is more personal and subjective, it accounts for the constructed knowledge of the market participant, and is more likely to have an impact on the real decision making activity.

The decision making model in the dealer's mind might admit several components: bid/ask/spread quotes, the dealer position (inventory and profit), the buyer/seller flow, and his/her expectation of the true price of the security, etc.. In this respect, the dealer faces a dynamic problem, in which requirements (market making) and resources (inventory and profit) are changing over time. Automating the decision support process (problem identification, developing alternative solutions, and selecting solutions), as proposed by Mintzberg (et al, 1976) and Talluru (et al, 1983), confronts several challenges. The dealer might find a difficulty in identifying or formulating his/her problem in a circumscribed way. The dealer has two roles to play that might be contradictory in nature: making a market (maintaining a flow of buyers and sellers) and maintaining a profitable position. The problem of making a market is vast and difficult to circumscribe in a concise way as it invokes tacit knowledge and personal experience of a particular dealer in a specific market. Although it is easy to calculate realized profit through a mathematical formulation, there is no recipe for profit making. Making profit in the market depends on the skill and experience of the

individual and on factors which are difficult to identify or judge, and where their contribution to profit is hard to assess. There might be no abstract or systematic solutions to ill-framed problems. As the problem of making a market and remaining profitable is not clearly formulated, it is hard to confirm that solutions respect the constraints of the problem. There is no point in delivering a solution for making profit to a dealer who cannot really quantify or formulate the level of profit to which he aspires, especially in adverse market conditions. There may be no explicit heuristic or algorithm for selecting a solution out of a set of ill-adapted solutions to a problem. The best amongst the worst solutions might look the most appealing.

The EM approach views decision support as a situated and experience based activity: It does not resort to full automation in the first instance, but favours semi-automation. Semi-automated decision support adapts to the evolving perception of the user. The aim of an EM model would be to complement and support the dealer's own conceptual modelling. Visualization plays an important role, and the dependencies between the components of the decision making model are numerous. Invoking Empirical Modelling technology in dealing with imprecise, qualitative problems in a way suitable for end-user development can support to a certain extent the decision making activity of the dealer. The dealer can conduct a what-if style of analysis or explore different scenarios by establishing or re-defining dependencies between the components of the model. Visualization can help in displaying the current state of the mental model of the dealer, and in making the state of the market transparent to the dealer in his own limited market view.

> ***Claim 4:*** ***EM technology supports an experience based approach to building an OFMM that can represent three kinds of agency in the financial trading. This can integrate cognitive support through open ended exploration of the finance domain and its corresponding trading process with operational support through depicting trading practice in specific markets. This might overcome the limitations of workflow description and structured methodological approaches in depicting the financial trading process and in exploring singularities arising in different trading practices in the financial market.***

Embodying the financial trading process model in an OFMM is important to support the understanding of the shift from the old to the new trading model. The questions to be

addressed are: Is a formal description of the financial trading process sufficient to explore the new trading model? Is there a general methodology for modelling the financial trading processes? Would the integration of the financial trading process in an OFMM help in giving better support to the decision making activity of market participants?

Modelling the financial trading process does not merely aim at depicting a workflow but should help to explore in an experiential way singularities arising in a particular trading process. A workflow description, such as the one found on the web page of chapter 6 in the thesis web page on the Thesis CD, depicting the financial trading process in a retail trade at NYSE, doesn't support the exploration of singularities arising in the trading process but simply serves a documentary purpose. Exploring the new trading model is better targeted in a situated financial trade process model. Such a model grows from experiential interaction with the model and the real financial market and helps in capturing a business state as experienced. This state evolves over time with increased domain knowledge and is associated with many different foci of attention [BRR00-1].

Representing the financial trading process by an abstract pattern[12] does not reflect the actual experiences involved in carrying out the financial trading process. Abstract computational states and business rules refer to interactions and situations that are presumed to be so well-understood that they can be detached from the real trading context and used reliably without the need to refer to the real financial market context. This abstraction separates experience in the real world from the financial trading process. Such a separation would be more readily tolerated in scientific and engineering contexts where theories and abstractions apply with a high degree of reliability. But in a social context this abstraction might detach the model from the real world domain and makes it useless in exploring different non-preconceived situations. From an EM perspective an OFMM should be provisional, subjective, and situated, and include both formal and informal knowledge of a domain. An OFMM could also be regarded as a model of the finance domain. It is related to its subject domain by a semantic relationship that is established experientially. The understanding of the finance domain model and the exploration of the new trading model proceed in parallel with the development of an OFMM. An OFMM is typically computer-based. Interactive experience with an OFMM and the finance domain is mediated by metaphors. A virtual reality technology can better mediate interactive experience when supported with EM principles for development.

[12] An abstract pattern may be described from a computational perspective by a formal state transition model (e.g. stateshart or PetriNet), and from a business perspective by a collection of rules that define the role of human participants [BRR00-1].

As a model of the finance domain, an OFMM can represent many kinds of agency in the financial market: open-ended, constrained, and circumscribed. Open-ended agency can be modelled by what is characterised in [Bey97] as a ***View 1*** agent: a group of observables with object-like integrity[13] unexplored potential to affect the model behaviour. Examples of entities that can be appropriately modelled by open-ended agency are traders, investors, financial indicators, electronic trading, payment, and transfer systems, economic indicators, financial events and news, political events and news, etc.. Such agency can be realized through role playing of participants (e.g. traders) and devices (e.g. trading systems) by human agents, and through direct manipulation of environmental entities (e.g. financial indicators). It may also be appropriate to introduce additional observables to reflect enriched observation of the agent. Constrained agency can be modelled by what is characterised in [Bey97] as a ***View 2*** agent: a ***View 1*** agent to which is attributed some patterns of stimulus-response and a role in changing state. Examples of entities that can be appropriately modelled by constrained agency are the different types of order placed by an investor to buy or sell financial instruments from the market (such as market, limit, market if touched, good till cancel, fill or kill orders), the different types of trading sessions adopted in stock exchanges (such as continuous or call market sessions), and the different types of execution systems (such as quote-driven dealer markets, order-driven markets, brokered markets, or hybrid markets). Such agency can be realized through introducing rules that can be executed manually or automatically at the discretion of the modeller. Circumscribed agency can be modelled by what is characterised in [Bey97] as a ***View 3*** agent: a ***View 2*** agent whose stimulus-response patterns are entirely predictable and that has preconceived roles in changing state. Modelling of this nature is only possible "at the point of circumscription" in the sense introduced in section 4.1.2. Such agency is represented in a model of a specific trading process in a specific exchange, such as the retail trading process in NYSE or NASDAQ.

Adopting a general methodology for modelling the financial trading process is hardly possible due to the situatedness of the social activity surrounding this process. Although a workflow description gives a high level view of the financial trading process special cases of situated and erroneous actions are not revealed.

Converting a financial trading process model into a software product for the financial market is not a straightforward task if the developer, designer and user are not involved in the initial

[13] In this context, object-like integrity refers to the coexistence of observables that is synchronized in time.

finance domain modelling and the conversion process. This can be largely attributed to the gap between the business and IT perspectives on modelling the finance domain [FRKBCJ00]. Participative process modelling is suggested in [CRB00] for the situated requirement engineering of an OFMM.

Integrating the financial trading process model with the financial analysis model in an OFMM can serve a better strategic decision support objective where the human market participant participates actively in the decision process.

> ***Claim 5:*** ***user-developer-designer collaboration enacted in distributed EM is essential in developing an OFMM. This can potentially deliver greater flexibility in the software system development activity of an application for the financial market derived from appropriate circumscription of an OFMM.***

Distributed interaction in an OFMM is needed to address three issues: user-developer-designer collaboration in the model development; group decision support; and participative modelling of the trading process.

The designer, the developer, and the user of an OFMM have a growing knowledge about key factors shaping the financial market. Traditionally, the user of a financial market model can apprehend the basic knowledge conveyed by that model, but cannot explore incremental knowledge related to the true price and the true profit determination. The same limitation holds true for the designer who might wish to incorporate new knowledge gained from the real world financial market domain. Incorporating new knowledge is a developer task that might necessitate a re-design of the model depending on the scale of the new insight to be recorded.

Encoding finance domain knowledge in a software application is not straightforward. This is mainly due to the gap between the designer's mental model and the developer's understanding of this model.

The designer's knowledge of the financial market and of trading mechanisms grows over time with the introduction of new systems and shifts from old to new trading models. In the context of financial markets, knowledge acquisition and construction is a continuous ongoing process and cannot be limited in space and time.

Distributed interaction enables the designer, the developer, and the user to cultivate their understanding of the requirement engineering of an OFMM and its use in developing a finished financial market product. Practically this can be achieved by providing privileged access to an OFMM script residing at a server that connects three clients: the modeller, the designer, and the user. Sun (et al, 1999) identified three possible modes of distributed interaction in a requirement engineering task (subordinative[14], coordinative[15], and collaborative[16]) and favoured the collaborative for an EM approach. In a collaborative relationship, there is no possibility of relying entirely upon closed-world representation and preconceived patterns of interaction. The interaction amongst all market participants is a situated intelligent interaction that can only be planned in advance to a limited degree, and knowledge for understanding emerges on-the-fly. Distributed interaction between the programmer, designer, and user of an OFMM allows data about requirements to be collected from all participants, makes it possible to visualize and analyse activities from the viewpoints of different participants, and provides open-ended interaction.

Every participant in the above model has their own domain knowledge: the designer has a deep knowledge in the finance domain; the programmer has a deep technical knowledge; and the user has an insight in doing a specific task that needs computer support and benefits from the designer knowledge. What makes the collaborative distributed interaction between these three participants feasible in an EM approach is the real meaning attached to observables and their state as communicated via a definitive script between the participants at the different client workstations. A sample collaborative interaction during the requirements engineering phase of an OFMM would address sharing insight between the designer, programmer, and user about the determination of the true price of the security. This is illustrated in the following dtkeden script communication between the three clients via the server.

```
sendClient("programmer"," true_price =
              (informedbuyers_per_unit_oftime == rush_rate) ?
               true_price+ticksize:true_price;");
sendClient("user"," true_price is (3*bid - ask)/2;");
sendClient("designer"," true_price is 64;");
```

[14] A subordinative distributed interaction assumes that the users should be able to provide all the knowledge required by designers because only they know what they want.

[15] A coordinative distributed interaction stresses the importance of user participation in design and postulates responsibilities for all the participants.

[16] A collaborative distributed interaction is concerned with sharing understanding that is socially distributed. Collaborators engage with issues of subjectivity and objectivity associated with distributed cognition [Hut95a, Hut95b] and common knowledge [Cro94, EM87].

Exploiting distributed interaction makes the OFMM more realistic and obviates the need to automate the role of different participants (e.g by having an intelligent dealer agent[17], or an intelligent investor agent). Smith (1997) argues that any approach to supporting a social context should take group activity into account. As introduced in chapter 2, a financial market comprises a social network involving individuals, companies, and governments. In the context of financial trading, this social network can be captured in the following diagram.

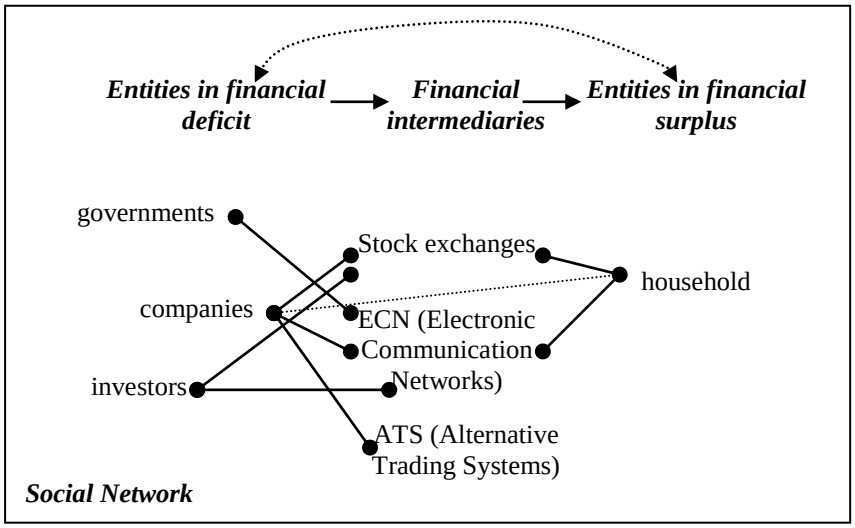


Figure 6.2 The social network in the financial market

Decision support is better targeted in a model where different human agents (buyers, sellers, and dealer) retain their identity and partake in their normal role as an agent within a distributed computing environment. This preserves the insight of each human agent into what action and judgement to take, while avoiding problematic issues associated with automating people.

A distributed model of the monopoly dealer simulation supporting the decision making of the dealer and of the investor would be more appropriate.

[17] This use of the term *agent* refers to a software component capable of autonomous actions. It differs from the use of the term agent in the EM literature, where it refers to an initiator of state change.

Claim 6: By adopting EM principles for developing an OFMM we can potentially have:

- a progressive computer-based support for the transition from the old to the new trading model;
- computer-based decision support for market participants;
- a closer integration between the software system development and financial trading related activities.

Supporting this claim by practical proofs is difficult in an academic research exercise conducted without industrial contact. It is helpful to review existing financial market models so as to get a better idea of what an OFMM can be and to choose a model that can be developed as a proof-of-concept for an OFMM. Such a proof-of-concept model can serve as a seed for more sophisticated OFMMs of the kind discussed earlier in this section.

6.3 The OFMM: technical implementation and practical applications

The technical implementation of an OFMM confronts the key issues associated with the shift from the old to the new trading model. These include:

- The experiential dimension of the computer-based software system development activity
- The openness, situatedness and evolutionary software system development activity
- The development of a computer-based artefact of subtle and dynamic nature capable of establishing a semantic relationship between the computer-based and real world activity
- The need to embrace emerging computer-based technologies (e.g. Virtual Reality, Web, multimedia, etc..)

6.3.1 The case study

The case study considered in this chapter draws upon several models for educational, analytical, decision making, and operational purposes that have been developed at academic and practitioner levels. These include:

The Wall Street Trader: The Wall Street trader is a wimp based application game with multimedia support, to help learning about the global financial community. The game has a simulated financial database covering two years of trading history. The user, the player of the game, is the investor and he/she can buy and sell stocks and read about a company's history. Some tools are available to the investor: an analyst (to help the user understand the news), an insider (to help the user uncover hard to find facts), and a spy (to see what other investors are up to). The game has a tutorial. The news database is fictional though inspired by real events. Each time the game is restarted the events are changing so that the game can be played many times without being repetitive.

Web-Based Trading Model: Boutell (1996) developed a web trading model using CGI (Common Gateway Interface) implementing a stock market trading system. As in online trading systems, the main features of this web-based model are to allow users (investors) to perform three tasks: 1) examine their portfolio; 2) buy and sell stocks; and 3) track the performance of stocks over time.

The Monopoly Dealer Simulation: This command line application simulates trading in a dealer market in which there is only one dealer (the user of the simulation model). The user's task (the sole dealer) is to set and adjust bid and ask quotes (raise, lower quotes, or narrow and widen the spread) to maximize his trading profits. The dealer (user) should know how to attract traders by adjusting his bid/ask quotes and spread. When the quoted bid/ask spread is wide, few uninformed traders will trade. To encourage uninformed traders to trade the spread should be narrowed. If the true security value is above the ask quotes, informed traders will buy from the dealer. They will sell to the dealer if the true security value is below his bid. Informed traders will trade more often and they will make larger trades when the dealer quotes are far from the true security value. The following diagram is relevant in depicting the buy/sell reaction of the informed investor according to bid/ask and true price value.

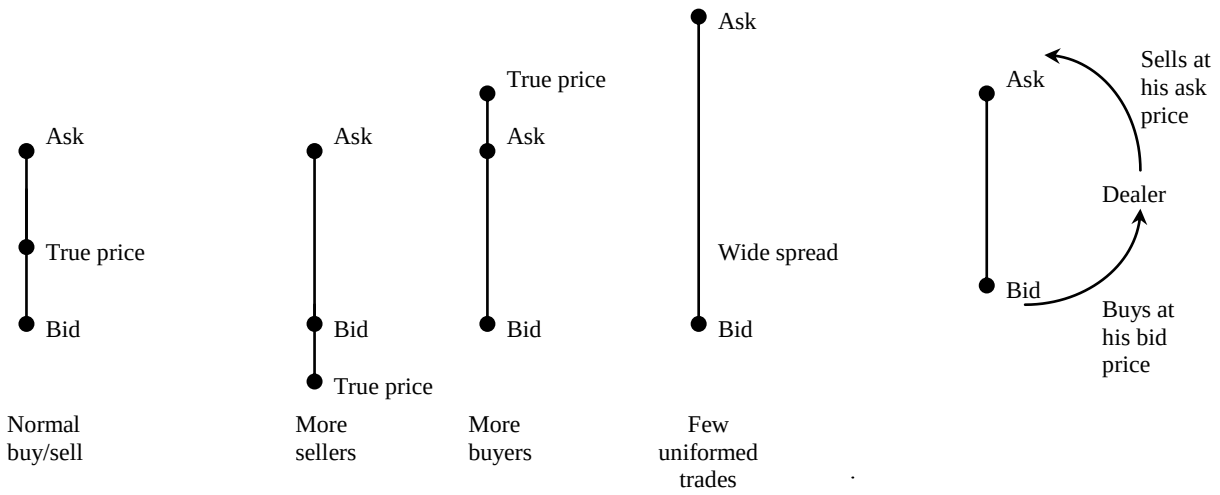


Figure 6.3 Investor's behaviour in response to dealer quotes

Head Trader: Head Trader[18] is a web-based educational simulation developed by the Nasdaq Stock Market®. It is based on an original model and software developed by two academics in the field of trading mechanisms - Robert A. Schwartz and Bruce W. Weber[19]. Head Trader simulates the experience of a Nasdaq Market Maker buying and selling stocks in a screen-based market environment.

Stock track: STOCK-TRAK[20] is an investment simulation, offering its users the opportunity to gain practical experience trading a wide range of investment vehicles.

(4i) System: the Integrated Investment Intermediary Information System (4i) developed by Consort securities systems Ltd.[21]. The system offers integrated investment management and stockbroking services. The system is WIMP based, with a GUI front-end and an SQL-Server relational database back end.

Konana et al (1999) model: In an attempt to model and analyse the effect of structural change in financial markets (the impact of online investment on the efficiency[22] of financial markets and the trading system in the short and long term horizon), Konana et al (1999)

[18] http://www.academic.nasdaq.com/headtrader/
[19] at the Zicklin School of Business at Baruch College, The City University of New York
[20] www.stocktrak.com.
[21] http://www.consort.co.uk
[22] In [KMB00], the processes determining market efficiency include: order flow, price discovery, and order execution. In that context, two types of market efficiency are identified: the efficiency perceived

developed models describing the investor choice process and the revenue flow and incurred cost in online trading are presented. The models are represented in a static diagram illustrating the workflow of a financial transaction.

The choice of a particular model as a basis for developing an OFMM should take into consideration important aspects in each of the financial market models / applications overviewed above. Our aspirations for the model to be developed are to convey important issues raised in finance in relation to studies in market efficiency, the determination of the true price of a security, the transaction cost, the determination of the bid ask spread, the impact of the behaviour of different agents in the financial market (investors, dealers, ..) on market performance and operation. The emphasis in this model will not only be on the represented knowledge but on how that knowledge is experientially constructed and conveyed to the user. Each of the financial market models reviewed above conveys basic knowledge about the financial market in some way. The Wall Street Trader, through its enhanced multimedia support, provides an entertaining tutorial on buying and selling stocks and analysing the market based on a limited database of financial indicators and news. The web-based application of Boutell (1996), gives some idea of the main features of an online trading environment, but does not tackle any fundamental financial issue. It is a good model from which to study the technical aspects of building online web trading environments. The model suggested by Konana et al (1999) for transaction cost in an online trading environment is quite inspiring, but cannot serve as a starting point for a computer case study. Models developed by practitioners are important and admit further development, but industry support is needed to get the model approved by different stock exchanges at an international level.
The author has chosen a simple model similar in spirit to Harris's simulation and the Nasdaq Head Trader as a basis for developing an OFMM that addresses various financial market issues and involves complex situations and cases. Our basic seed model considers a market where a single security is traded and the dealer adjusts his quote to attract various types of buyer and seller.

6.3.2 The OFMM: A proof-of-concept

Two models were developed as a proof-of-concept of the practical use and technical implementation of an OFMM: using EM technology, and using virtual reality technology.

---

by investors, and the real efficiency of the transactions and the patterns of information flows beyond the interface.

6.3.2.1 Model developed using EM technology

As introduced in chapter 3, the principles of the Empirical Modelling approach are based on the concepts of observation, agency and dependency. The initial analysis of a domain to be modelled is made by identifying *observables* considered relevant by the modeller. These observables are grouped around the *agents* regarded as sources of change in those observables. The *dependencies* between observables are expressed in *definitions*. A set of definitions – a definitive *script* – corresponds to a single *state* of the model. Any particular state of the model should directly correspond to a possible state of its *external* referent. All these identifications (observables, dependencies, and agents) are provisional and subjective: they represent the viewpoint of the modeller. The *techniques* of the EM approach involve an analysis that is concerned with explaining a situation with reference to agency and dependency, and the construction of a complementary computer artefact – an *interactive situation model (ISM)* – that metaphorically represents the agency and dependency identified in this process of construing. There is no preconceived systematic process that is followed in analyzing and constructing an associated ISM. The modelling activity is open-ended in character. The LSD notation is used to classify observables and dependency in agent interaction.

The construction of the ISM for the chosen model involves an agent oriented analysis where the following observables are recorded:

- dealer (bid, ask, spread, inventory, true profit, actual profit)
- buyer/seller (informed / uninformed, quantity bought/sold, transaction price)
- simulation clock (simulation time and speed)
- transaction details (side, quantity, actual price, true price)
- security type
- market (monopoly dealer market, single security)
- warning messages to dealer

Observables are grouped around *agents* that are regarded as sources of change to those observables. Observables associated with agents are classified as oracles, handles, and derivates. An LSD template for the dealer agent takes the following form:

```
Agent Dealer {
State
    inventory, bid , ask , spread, actual profit, buyers/sellers flow,
    current status and history of transactions, time clock, his estimated
    true value of the security
Oracles
    flow of orders, order side (buy/sell), order quantity, inventory level,
    actual profit, his estimated true value of security, his knowledge of
    trader type (informed/uninformed)
Handles
       Bid, ask, spread
Protocols
    if (estimated true price > ask) || (informed trader rush to buy) ) →
    raise ask
    if (estimated true price < bid) || (informed trader rush to sell)) →
    raise bid
    if (spread is wide)|| (few uninformed traders are trading) → narrow the
    spread
    if (inventory is approaching the limit of +/-10,000 ) → adjust quotes
    to attract buy and sell orders appropriately
}
```

The ISM of the considered model is developed using the client server architecture of dtkeden (cf. Figure 6.4). The server provides a global market view including the knowledge hidden from market participants (such as the true price, the true position of the dealer, the type of an investor – informed/uninformed), as well as the publicly known information such as the dealer bid/ask/spread quotes, the current status and history of transactions. The client provides a dealer's view that includes the observables that the dealer can view (oracles), such as his position (actual profit and inventory level), the flow of buyers and sellers, and the current status and history of transactions. The dealer's actions (raising/lowering quotes) are also undertaken via the dealer's view and the results of these actions are transmitted to the server. Agents in the model are the dealer, the investor (buyer/seller), and the clock.

Many issues surrounding the trading behaviour of buyers and sellers and the strategic decisions of a dealer (the true price of the security; the role of the dealer in the determination of the true price, and transaction price; the type of investor (informed/uninformed); the buyer/seller and transaction flows; and the hidden intentions of the investors to buy or sell) motivate the thinking of different construals, each reflecting a particular scenario, to account for our weakly structured knowledge of the complex trading system. For instance, it might be that the true price is determined by the trading pattern, or that it is influenced in a non-

deterministic manner by external events. Rules to govern the interaction are imposed to reflect each possible explanation of the state of observables.

Different construals may enable the exploration of issues concerned with market efficiency[23], trading behaviour, timing considerations, and transparency. Rules are imposed to govern the interaction with the construals and to reflect one possible explanation of the state of observables. For example, in the construal reflecting a trading behaviour, we impose a rule on the interaction with the model stating that the trading behaviour (buyer/seller flow) responds to dealer quotes and to the value of these quotes relative to the true price (cf table 6.1). This assumes that the trading pattern is totally under the control of the dealer when the true price is fixed and that informed investors are trading. Market efficiency is a more advanced concept in financial analysis and is yet more complicated to construe in a computer-based artefact. It can be explored by imposing rules about how long a trading pattern takes to affect true price, and consequently dealer actions and vice versa. Time is also an important factor in governing trading patterns and dealer actions, as there may be a delay before trading intentions materialize into trading actions. Market transparency is also an important concern: a simple interpretation to transparency would involve making the true price visible to all trading agents including the dealer. The construals should also take into consideration the conflict of interest faced by a dealer when aiming at maintaining an ordered market flow and making high profit.

Trading behavioural impact
Dealer actions ➔ Trading patterns
External events ➔True price ➔Trading patterns
Trading patterns ➔ dealer actions ➔ trading patterns
Trading patterns ➔ dealer actions ➔ true price ➔ trading patterns
Market efficiency impact
Dealer actions ➔ trading patterns ➔ true price
Dealer action & external events ➔ trading patterns ➔ true price
Time duration impact
Duration for trading intention to materialize in actual trading action ➔ true price ➔ dealer actions
Market transparency impact
True price is transparent to dealer ➔ (true price ➔ dealer behaviour)
True price is transparent to trader ➔ (true price ➔ trading behaviour)
note: the sign ➔ means indicates a direct impact on

Table 6.1 Different construals in the case study

[23] This refers to efficiency as described in the efficient market hypothesis (EMH).

In modelling state in the financial market and the complexity of the trading activity an ISM is a rich computer-based artefact that can capture our evolving knowledge of traders' behaviour in terms of various observables representing:

- The probabilities of buying and selling,
- The potential time of delay for decision-making of buyers and sellers,
- parameters to determine density of trade
- how many buyers and sellers there are around
- how long are buyers and sellers prepared to wait
- how much buyers and sellers are influenced by the spread
- how the behaviour of buyer and sellers is correlated to true price

The model can include parameters and potential dependencies to support the rich construals reflecting behaviour of traders and the complexity of the trading activity. The state of the observables in the model can be changed through new definitions and re-definitions.

The ISM helps in exploring experientially fundamental and subtle issues related to the true price of a security and its essential elusive nature and relationship to the trading patterns and trader agency. The true price is a typical example of something that is not represented most effectively by a numerical value, as its meaning is arbitrated by agency and interaction. Relevant agents influencing the state of the true price are buyers, sellers and dealers, as well as external factors such as supply and demand The true price can be construed in various ways:

- as that value about which a dealer can most profitably pitch their bid and ask prices so as to maximise trading throughput
- as influenced by supply and demand
- as expressed through responses of buyers and sellers to the situation
- as influenced by the roles of informed and uninformed buyers
- as sensitive to potential dealer's influence and their motivation for being able to "estimate true price"
- as sensitive to external factors beyond the control of buyers and sellers

Modelling true price in a tkeden model in principle allows us to exploit two powerful features: (1) construals that can be manipulated interactively and dynamically; and (2) dramatisation of the roles of many independent agents.

The ISM for a financial trading context could be adapted to a sale shop context. Different scenarios that arise in a sale shop can be considered where the dealer can gauge the level of interest of buyers. An LSD description frames the pattern of agency and dependency in the model, and an ISM serves as a medium to experientially explore such a description.

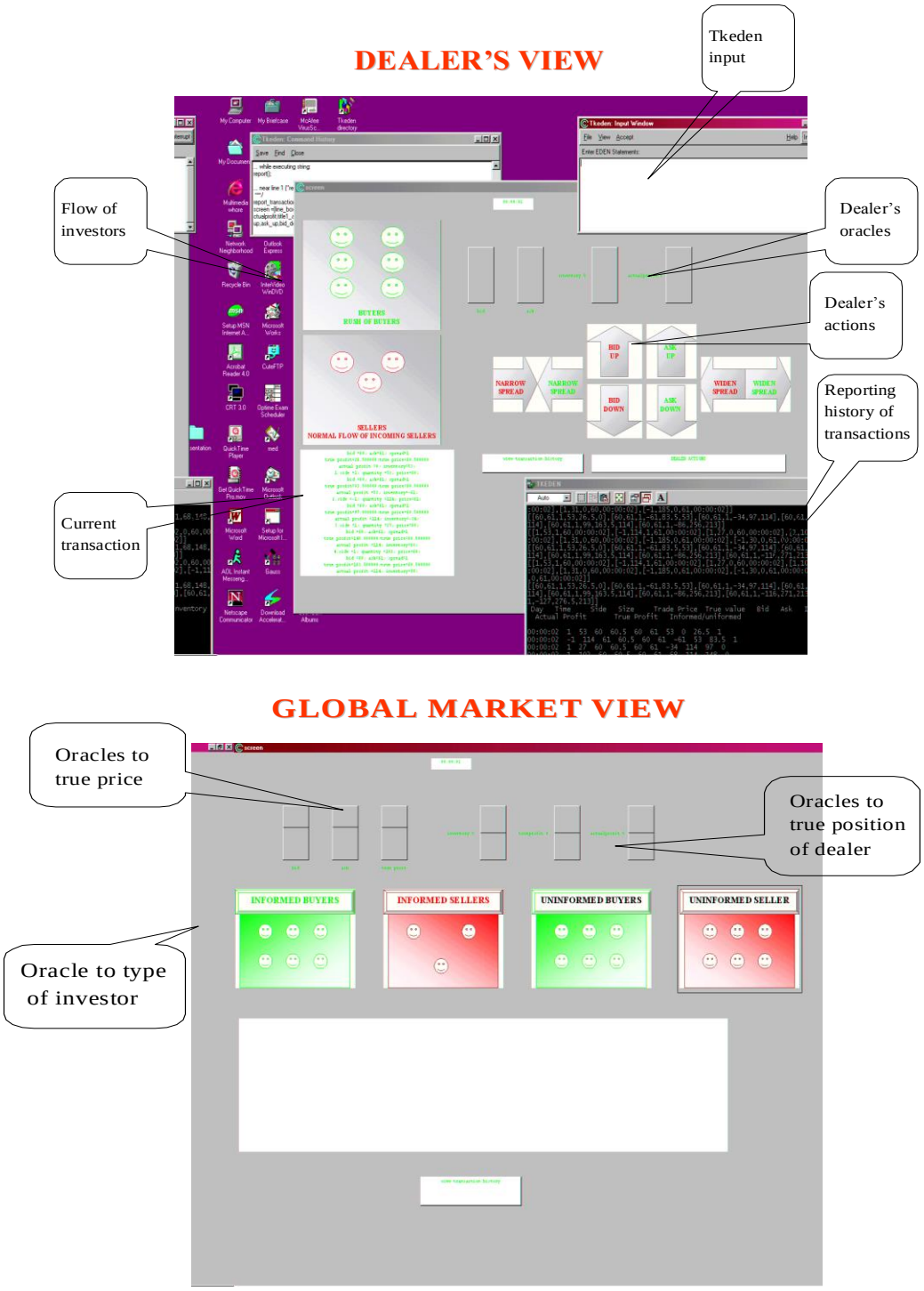
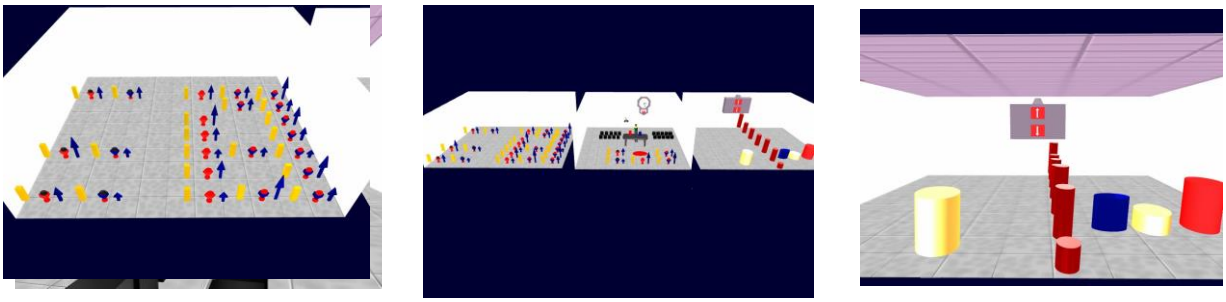


Figure 6.4 the distributed monopoly dealer simulation

6.3.2.2 Model developed using VR technology

The EM model for the considered case study can serve two purposes: as an artefact in its own right, and as a requirements model. In constructing a VR scene for the case study, the EM model serves as the requirements analysis. As an additional exercise, we have to find an appropriate visualization for abstract numeric indicators, agent actions, and the human (user) role in the scene, and to add sound support to produce warning messages to the dealer. The VR scene includes 3 rooms: the dealer action room, the transactions history room, and the hidden knowledge room. Transactions are saved in a file and are visualized in the transaction history room. The set up for the experiment is developed on a Silicon Graphics Machine running Irix6.5, using Parametric Technology Corporation's VR modelling tool Dvise, and the peripherals includes a 3D mouse as an input device, CrystalEYES glasses for the Stereographic image and 3D auditory feedback. Figure 6.5 shows snapshots of the VR scene.



Dealer Actions View *VR Scene of the Monopoly Dealer* *Transaction History View*

Figure 6.5 The VR scene

There is a very significant distinction between VR modelling for areas such as robotics as represented in papers such as [GC00], and its application to Virtual Trading. Whilst we can reasonably speak of "using VR to model the reality of a manufacturing assembly process", the reality of the virtual trading environment is an altogether more elusive concept. Where manufacturing assembly deals with objects and actions whose objectivity and real-world authenticity is uncontroversial, virtual trading is a prime example of an activity in which the impact of technology upon human cognition is prominent, and the character of its agencies and observables is accordingly hard to capture in objective terms.

The challenges faced in the use of VR for constructing virtual environments for financial trading are best revealed by drawing a comparison with its use in computer-aided assembly [GC00]. This comparison reveals a difference in the objective, considerations, approaches, and user role in constructing VR scenes for different contexts.

- The main objective in using VR for virtual trading is enhanced cognition of financial market phenomena; in the case of virtual assembly the main objective is to minimise the need for building physical prototypes.
- The issues to be considered in applying VR in financial markets and in virtual assembly differ in nature and importance. In virtual assembly, the major concerns are proper 3D picture capturing, conversion, and adding behaviour to objects; in VR for financial trading, they are geometric abstraction of financial concepts, integration with financial database, and distributed interaction.
- The steps followed to create a VR scene for virtual assembly and for financial markets are different. A linear, preconceived, set of processes can be followed to develop a VR scene for virtual assembly. These can be framed in three stages: defining objects to be assembled, preparing the assembly geometry for visualisation, and adding behaviour to visualised objects. Creating a VR scene for a financial trading context is more complicated and cannot be framed adequately in a pre-conceived way. However, a broad outline can be traced to guide the VR construction process. This involves: identifying entities (both those that admit geometric abstraction and those that have already a well recognised geometric representation) to be included in the VR scene; choosing an appropriate geometric representation for these entities; adding a situated behaviour and visualisation to entities; identifying the external resources (such as databases, files, data feeds, etc.) to be interfaced to the VR scene; and framing the role of the user intervention in the simulation.
- Where human intervention is concerned, the user's role in the VR scene is more open-ended in a financial context than in an assembly context. In a VR scene for assembly the immersion of the user is very important. Armed with helmet, gloves, and three-dimensional pointing device (such as 3D mouse and keyboard), the user can manipulate virtual objects with his hands. The user's hands, guided by the user's brain, interact directly with virtual objects. This makes virtual reality environments more appropriate for the assembly task than any alternative technology. Construing financial market phenomena is a function performed by the human brain. The mental model of the designer can be abstracted in a static diagram, a 2D computer artefact, or a VR scene. Geometric objects in the virtual scene might admit no counterpart in the real world - they are purely geometric metaphors. This makes a virtual scene just one of several possible representations. It also motivates a prior situated analysis exploring possible construals pertaining to the social context.

The above comparison highlights the need to support VR technology with principles and techniques to analyse and construe social contexts and to adopt appropriate visualisations for abstract entities (such as financial indicators) that have no real geometric counterpart. Current technologies for Empirical Modelling can help in construing financial situations and in representing state and the analysis of agency in state change, whilst VR offers enhanced visualisation and scope for user immersion and experience of state.

A VR scene can help in exploring a particular state in a social context. In developing the VR scene of the considered case study, the following conclusions were drawn:

- The pre-construction phase for a VR scene can benefit greatly from concepts drawn from the Empirical Modelling literature such as modelling state, state change, and the initiators of state change.
- VR technology needs to be better adapted for the representation of multiple agents acting to change the state and corresponding visualisation in a VR scene.
- The successful application of VR technology in modelling a social and data intensive environment will rely upon integrating VR with other programming paradigms such as databases and definitive programming.

6.4 Towards a Broad Foundation of Computing: a proposed EM – VR Merge

The VR model of the financial market, considered in this chapter, motivates the proposal of an EM-VR merge to empower VR technology with basic principles and techniques for modelling social contexts. This contributes to a broad foundation of computing based on EM and aiming at merging, where possible, disparate technologies.

Maad et al (2000) investigated the applications and prospects of VR technology in supporting the development of virtual environments for financial trading. The research suggests future prospects for VR technology in developing virtual environments for financial trading when integrated with other technologies such as database, and EM technology. A new perspective on the design, application and use of VR technology in modelling a social context, such as the financial trading one, is motivated. This takes into account the central human role, the need for appropriate geometric abstractions of concepts and real entities, the large amount of

corresponding data, and the identification of actions and behaviour associated with virtual objects.

Further interdisciplinary research in the areas of financial market microstructure, electronic financial trading, Virtual Reality technology, database technology, business process modelling, interface design, and Human Computer Interaction is to be undertaken. The aim is to identify the role and uses of VR technology in developing virtual environments for financial trading and the adoption of appropriate performance metrics to assess its added value[24]. This involves the exploration of the use of VR technology in modelling the financial trading process, supporting the financial decision making activity, and designing interfaces for virtual environments for financial trading.

6.5 Summary and future outlook

Modelling financial markets is a broad and ambitious objective. Financial markets are facing major structural changes due to increased competition and the rising demand and need for cross-border trading. There is a rising need for better models to depict the reality of financial markets, and to explore the shift from the old to the new trading model. Interaction in a trading environment is particularly subtle and complex because it combines real-world knowledge and observation with real-time interpretation of abstract numerical data and indicators. Traditional mathematical models are not sufficient for such applications, where human behaviour is of paramount importance.

Modelling the impact on financial market structures of the introduction of new technical trading systems in general, and online trading in particular, needs more than traditional econometric approaches that operate on past transactions data. The ability to forecast the impact of a new technology or structural change prior to its introduction, and to investigate an optimal financial market structure and organization needs more than traditional financial modelling techniques and calls for cognitive and experience based modelling.

In assessing the prospect of EM technology in modelling financial market, an Open Financial Market Model is proposed. The theoretical and practical framework for developing the OFMM relies on EM principles and foundations. An OFMM differs from a program or a finished software product in its degree of openness, in its customisation for use by academics

[24] [MBG01] suggests cognitive dimensions [Gre00] to assess the potential benefits of VR modelling in a social context.

and practitioners, and in its representation of the real financial market domain. A brief summary of the key characteristics of an OFMM follows:

- An OFMM aims at exploring the new trading model
- By adopting an EM approach, an OFMM serves as a support for situated knowledge construction of key concepts in the financial market such as trading cost, the determination of the true price of a security, price movement, and behaviour of market participants.
- The use of EM technology in an OFMM supports semi-automated decision making
- Distributed interaction features in EM technology enable user, developer, designer collaboration in the development of an OFMM.
- Distributed interaction involving market participants (broker, dealer, investment, trading systems) in an OFMM supports the exploration of a trading process model and strategic decision support to market participants.

EM proposes basic principles for software system development in the financial market context and supports the claim in [Sun99] that software system development is a context dependent activity that cannot be detached from its real world context and cannot be abstracted in formal methodologies and rigid development cycles.

Chapter 7

Distributed Modelling of Financial Instruments: An EM Approach

7.0 Overview

This chapter considers distributed support for financial instruments modelling. It argues that an adequate support for modelling financial instruments should take into account the interaction between human participants in the modelling activity. Section 7.1 considers the case study of an affine interest rate model developed by N. Webber[1] using a spreadsheet application. Section 7.2 overviews technology support for the considered case study: web-based support and distributed Empirical Modelling. Section 7.3 discusses the broad implications of distributed EM technology on modelling financial instruments. The chapter concludes with the prospects of Empirical Modelling technology applying in developing distributed environments for modelling financial instruments.

[1] A senior lecturer at Warwick Business School

7.1 The Case Study

The case study considered in this chapter is an Excel spreadsheet model developed by Nick Webber for an affine interest rate model detailed in [JW00]. This case study is selected for its simplicity (as a financial model) and because of the importance of interest rate models in pricing financial instruments. The original spreadsheet model was introduced for classroom use during a one week workshop on computational finance for MSc students in economics and finance at Warwick Business School. The use of the model by practitioners is discussed in this chapter, but the research work relating to this case study was conducted without any contact with industry.

7.1.1 About Affine interest rate models

Interest rate is often referred to as the time value of money - a dollar today is worth more in one-year's time because you can invest it and earn interest on it. Interest rates are associated with all borrowing activities, which are vital to the growth of the economy. However, interest rates are subject to fluctuation over time, which poses a risk for borrowers and encourages them to resort to hedging[2] strategies [JW00]. Moreover, new financial instruments based on interest rate have recently been engineered (e.g. interest rate option, swaps, etc). Their present valuations are computed, and their marked to market valuations[3] are tracked, thus widening the range of products in the interest rate market. This, in turn, influences the shape of the yield curve[4] in different markets as well as the efficiency[5] of these markets.

In the face of these recent developments in the interest rate market, models have been built to price and hedge interest rate instruments, to conduct risk management, and to better understand interest rate movement [JW00]. Interest rate models provide a quantitative framework for describing interest rate movements and valuing and hedging interest rate products. By fitting a model to available interest rate data, we can discover the dynamics of interest rates and the way that interest rate and derivative prices are related. This helps us in better understanding, pricing, and hedging interest rate products. James (et al, 2000) summarize the current status of interest rate models as follows: "although it is desirable to use

[2] Hedging a deal involves undertaking a similar or identical offsetting deal whose cashflow is opposed to that of the first deal [JW00].

[3] The marked to market value of a financial instrument or portfolio is its value when it is benchmarked to current market prices [JW00].

[4] The yield curve or term structure of interest rates is the set of interest rates for different investment periods or maturities [JW00].

models to predict future movement in prices or rates, it is somewhat unfortunate to realise that no interest rate model comes close to achieving this goal. Despite the enormous amount of research in banks and universities that has gone into interest rates, there is no reliable method of accurately predicting tomorrow or next week's interest rates. Interest rates models cannot be used to accurately predict daily ups and down, they can merely describe distributional properties of interest rate movements. Interest rate models are generally statistically based. Valuing and hedging financial instruments brings liquidity to the market enabling the financial industry to operate."

The finance literature describes a wide range of interest rate models based on different techniques, suitable for different purposes, and working with specific types of data sets. This variety of interest rate models makes it difficult to set comparative benchmarks. This is partly because the data that is available to test interest rate models is of poor quality, and because of the huge variability in market conditions. The priority when testing and using interest rate models is pricing and hedging. In this process the model is fitted to available current market data. Its floating parameters are adjusted until the model prices for various liquid market instruments match those seen in the market. It is essential that models give accurate prices for liquid market instruments. James (et al, 2000) put interest rate models into four categories: affine models, whole yield curve models, market models and price kernel models.

An interest rate model [JW00] is defined by its state variables and by their processes. The values taken by the set of state variables in a model completely determine the state of the system. The state variables processes determined how they change through time.

All stock prices and interest rate processes are stochastic processes. They change randomly over time, but the manner in which they change can be modelled. It is possible to divide the change in their values into two parts, the first is a non-random, deterministic component, called the drift of the process, and second is a noise term – the random part, called the volatility component of the process. For financial time series without jumps, noise is assumed to be a function of a Wiener process[6].

My case study focuses on the implementation of the Longstaff and Schwartz Two-factors affine[7] term structure model originally developed by N. Webber using a spreadsheet

[5] Market efficiency [6&7] is determined by the ability of prices to reflect present, past, public, and private information.

[6] a certain standardized stochastic process

[7] Affine interest rate models are the most popular models in the financial industry due to their tractability and flexibility. Affine models are widely studied in the finance literature, and are defined in terms of abstract state variables. They are separated into three main types Gaussian affine Models, CIR affine models, and three-factor affine family.

application. In affine models, spot rates are affine functions of the state variables, and the state variables can be taken to be spot rates of particular maturities. To ensure that these properties hold, the dynamics of the state variables is heavily restricted. The short rate is affine in the state variables.

The Longstaff and Schwartz Two-factors affine [JW00] model has two underlying state variables x_t and y_t . The processes for x_t and y_t are:

$$dx_t = (a - bx_t)dt + c\sqrt{x_t}\,dz_{1,t}$$

$$dy_t = (d - ey_t)dt + f\sqrt{y_t}\,dz_{2,t}$$

The short rate r_t is a linear function of x_t and y_t, and the short rate volatility v_t defined by $dr_t\,dr_t = v_t dt$ also turns out to be a linear function of x_t and y_t

$$r_t = \alpha x_t + \beta y_t$$

$$v_t = \alpha^2 x_t + \beta^2 y_t$$

For constants α and β the two previous equations can be inverted and the processes for r_t and v_t written down in terms of each other

$$dr_t = ((\alpha\gamma + \beta\eta) - \frac{\beta\delta - \alpha\xi}{\beta - \alpha} r_t - \frac{\xi - \delta}{\beta - \alpha} v_t)dt + (\frac{\alpha\beta r_t - \alpha v_t}{\beta - \alpha})^{\frac{1}{2}} dz_{1,t} + (\frac{\beta v_t - \alpha\beta r_t}{\beta - \alpha})^{\frac{1}{2}} dz_{2,t}$$

$$dv_t = ((\alpha^2\gamma + \beta^2\eta) - \frac{\alpha\beta(\delta - \xi)}{\beta - \alpha} r_t - \frac{\beta\xi - \alpha\delta}{\beta - \alpha} v_t)dt + (\frac{\alpha^3\beta r_t - \alpha^3 v_t}{\beta - \alpha})^{\frac{1}{2}} dz_{1,t} + (\frac{\beta^3 v_t - \alpha\beta^3 r_t}{\beta - \alpha})^{\frac{1}{2}} dz_{2,t}$$

For certain γ, δ, η and ξ, the Longstaff and Schwartz model can be solved for the prices of pure discount bonds. In terms of r_t and v_t, pure discount bond prices are:

where τ=T-t and

$$A(\tau) = \frac{2\varphi}{(\delta + \varphi)(e^{\varphi\tau} - 1) + 2\varphi}$$

$$B(\tau) = \frac{2\varphi}{(\upsilon + \varphi)(e^{\varphi\tau} - 1) + 2\psi}$$

$$C(\tau) = \frac{\alpha\varphi(e^{\psi\tau} - 1)B(\tau) - B\psi(e^{\varphi\tau} - 1)A(\tau)}{\varphi\psi(\beta - \alpha)}$$

$$D(\tau) = \frac{-\varphi(e^{\psi\tau} - 1)B(\tau) + \psi(e^{\varphi\tau} - 1)A(\tau)}{\varphi\psi(\beta - \alpha)}$$

$\upsilon=\lambda+\xi$, $\varphi=\text{sqrt}(2\alpha+\delta^2)$, $\psi=\text{sqrt}(2\beta+\upsilon^2)$, and $\kappa=\gamma(\delta+\varphi)+\eta(\upsilon+\psi)$
and λ is a market price of risk.

The process of fitting an interest rate model to historical and current market data is known as estimation. It is also called calibrating a model if the parameters of the models are being estimated from current market data [JW00]. All yield curves models have a number of parameters that are floating (they may be adjusted until the model fits the available data within reasonable limits).
Nelson and Siegal curves are used to fit yield curves non-parametrically. The Nelson and Siegal curve is:

$$f_0(\tau) = \beta_0 + (\beta_1 + \beta_2\tau)e^{-\kappa\tau}$$

Nelson and Siegal curves can be used to model either the forward rate curve or the spot rate curve. The spot rate curve is derived as:

$$r(\tau) = \beta_0 + (\beta_1 + \frac{\beta_2}{\kappa})\frac{1 - e^{-\kappa\tau}}{\kappa\tau} - \frac{\beta_2}{\kappa}e^{-\kappa\tau}$$

The implementation of Longstaff and Schwartz and Nelson and Siegal models (calibration, documentation, extension, re-use, and result reporting) is discussed in the following sections.

7.1.2 The spreadsheet model implementation

The spreadsheet developed by Nick Webber, implements the Longstaff and Schwartz and Nelson and Siegal models. The Longstaff and Schwartz model is calibrated to fit term structure data generated by the Nelson and Siegal model. No real term structure data (interest rate against maturity) is used. The Nelson and Siegal model term structure simulates the real data. Parameters of both models can be varied. The success of the fitting of the Longstaff and Schwartz term structure data with that of Nelson and Siegal term structure data (that simulate real term structure data) is assessed by monitoring the root-mean square difference between the data of the two models. Curves for the term structure data generated by the two models are plotted, and parameters are varied until the two curves nearly fit each other. The following picture is a snapshot of the term structure fitting model.

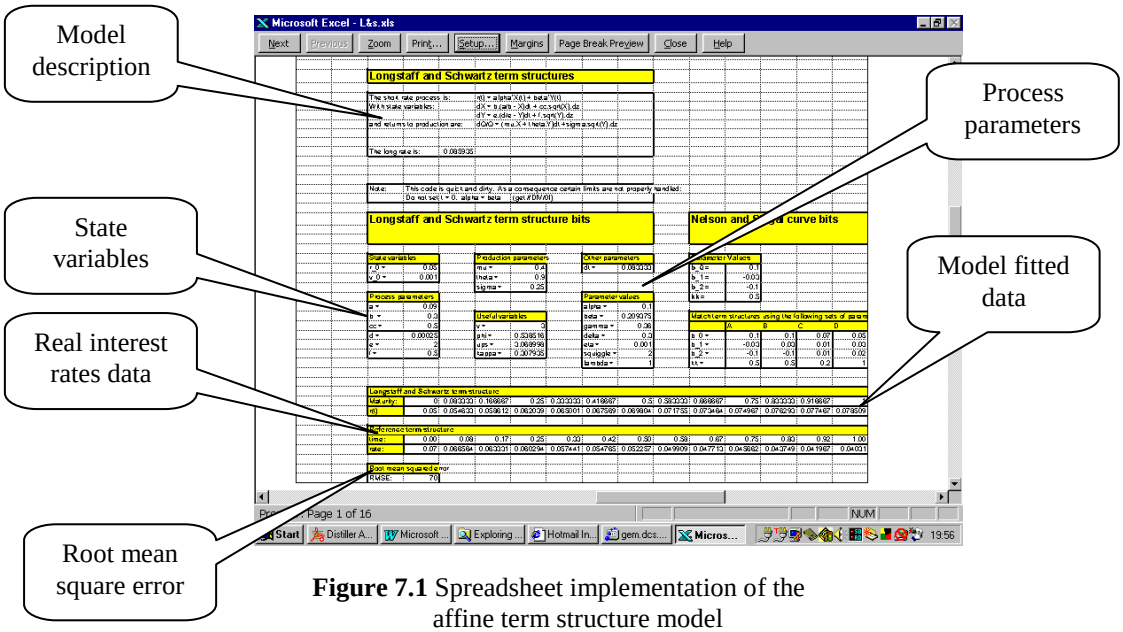


Figure 7.1 Spreadsheet implementation of the affine term structure model

The spreadsheet in Figure 7.1 is used to implement the two models. It clearly shows two parts: one part related to the implementation of the Longstaff and Schwartz model, and the other part related to the Nelson and Siegal model. The models are used to generate two interest rate series r(t) for the same set of generated maturities. Maturities are generated in consecutive spreadsheet cells starting from zero and incrementing by dt for the next cell (dt is a parameter set by the user). The spreadsheet gives a brief description of the Longstaff and Schwartz model in terms of state variables and process parameters and returns production process. The state variables and process parameters of the Longstaff and Schwartz model are

each located in a spreadsheet cell. Also, the parameters of the Nelson and Siegal curve (used to simulate real term structure data) are located in their own spreadsheet cells. The figures below show a plot of the Longstaff and Schwartz and the Nelson and Siegal term structure data, and the macros used to implement the Longstaff and Schwartz model.

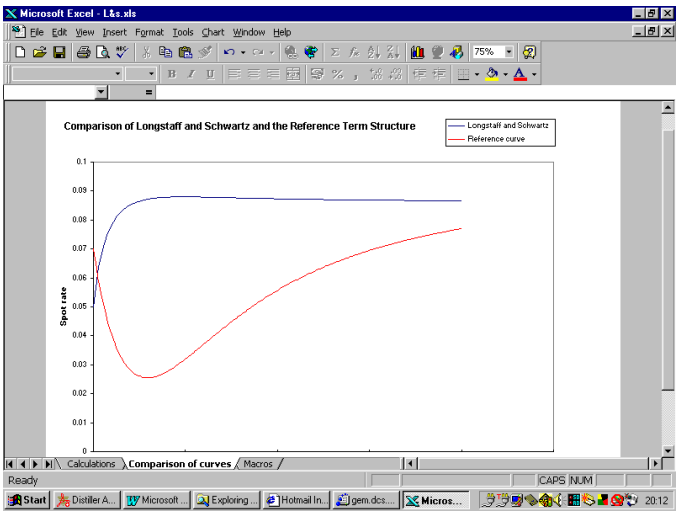


Figure 7.2 Comparative graph of the real and fitted term structure data

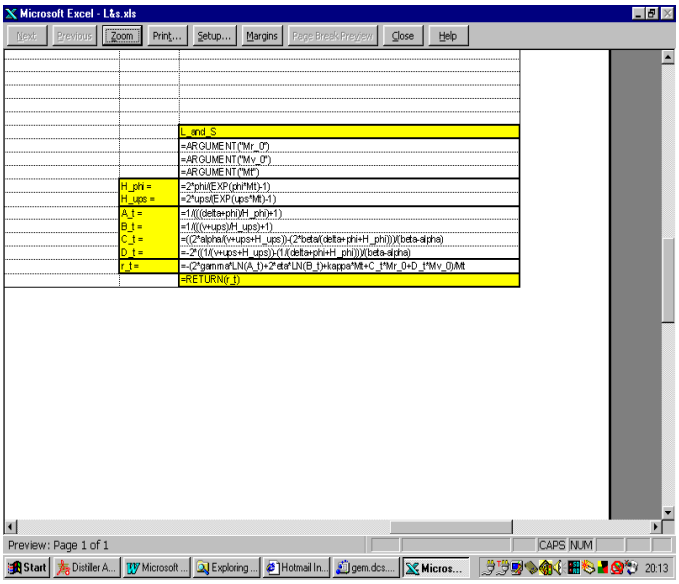


Figure 7.3 Macros used in the model

7.1.3 Classroom Observation

The spreadsheet described above was introduced in the computer laboratory for a one week workshop of computational finance given to MSc students in economics and finance at Warwick Business School.

In the computer lab, the students were supposed to calibrate the parameters of the affine interest rate model to fit Nelson and Siegal term structure data (simulating real data). The spreadsheet model estimates the error between the fitted data and the simulated real data. The supervisor introduced the spreadsheet to students by describing it on the blackboard. The students, each sitting on separate computer workstations, started varying the parameters of the model to get the best fit. They used direct face to face collaboration to discuss their results and share understanding of the model.

Some problems were encountered by students when working with the spreadsheet implementation of the two models. This was mainly due to the lack of a medium to store the result of their trials, to share the understanding of the model, and to report the results not only of the latest saved calibration of the model but of the history of their work. Sometimes good values of trial parameters were subsequently replaced by poor ones. These problems subverted the learning objective of the spreadsheet model and tended to subvert students' understanding of the relationship between the model parameters and the fitting error.

7.1.4 The use by practitioners

In a discussion with Nick Webber about the use of the considered model by practitioners to help in the hedging and pricing of financial instruments activity, the following static workflow model emerged (cf. Figure 7.4). This documentary workflow model describes the sharing of an interest rate model among the different departments of a financial institution. The sales department captures the specification of a financial product requested by a client. The term financial product used in this context refers to a financial contract (a hedging instrument, a stock, a bond, a derivative instrument). The sales department communicates the client's request to the quantitative analysis, the trading, and the risk and measurement department. These departments share and discuss different interest rate models to come up finally with the product requested by the client. Once developed, the financial product is delivered to the customer. This workflow description is not tailored to a particular financial institution. It just gives a general idea of the business process model used to develop a financial product and deliver it to a customer. At the heart of this business process model are financial instrument models similar to the ones considered in the case study.

The development of a financial product resembles to some extent the development of an engineering product. Financial experts share their knowledge to craft a product customized to the client's needs.

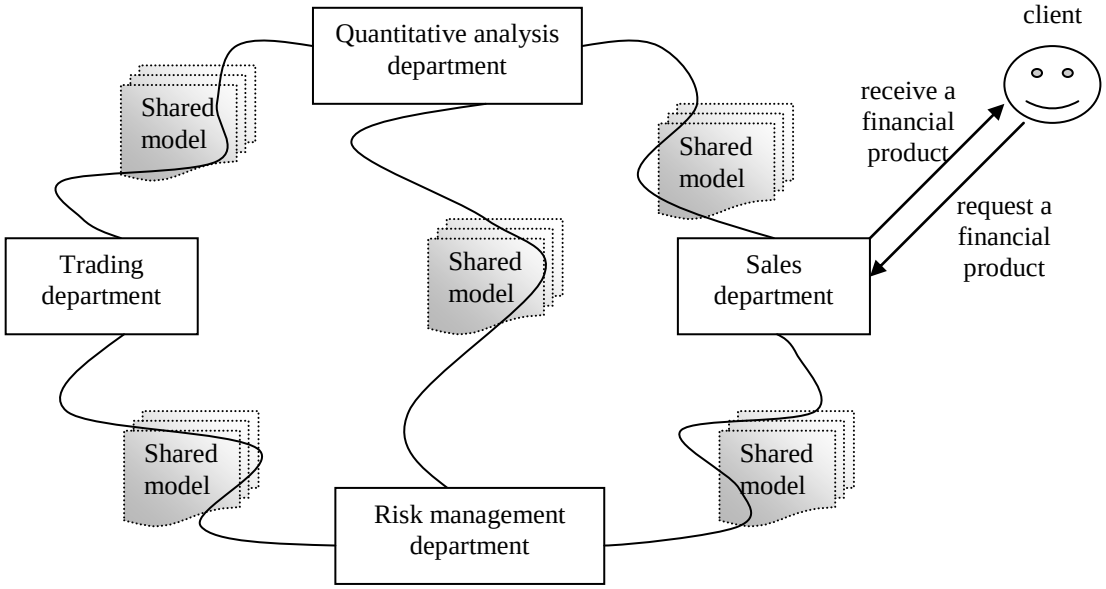


Figure 7.4 The flow across the financial enterprise of a shared model of a financial product requested by a client

7.2 Re-engineering The Spreadsheet Model

This section presents two models developed by the author for the case study considered: a web-based model, and an EM model.

Note that the analysis in this section does not tackle financial issues related to the development of the affine interest rate model but is confined to discussing the computer-based implementation.

7.2.1 The web-based model

The first viable alternative to a single user spreadsheet model is a web based model. The author developed a web based model using (CGI) Common Gateway Interface programming written in C with the aid of some web-based libraries: CGIC (common gateway interface C library) and GD (a graphics library).

Screen shots of the web pages of the model are depicted in the figure below. The importance of the web based implementation is the rich documentation that can be supplied with the model and that can assist the user in understanding the model. The web model consists of a fill-in form to calibrate the parameters of the model. Once the user clicks on the submit button a common gateway application is invoked to evaluate the term structure data of the Longstaff and Schwartz and the Nelson and Siegal models and to estimate the root-mean-square difference between the term structures data series generated from the two models.

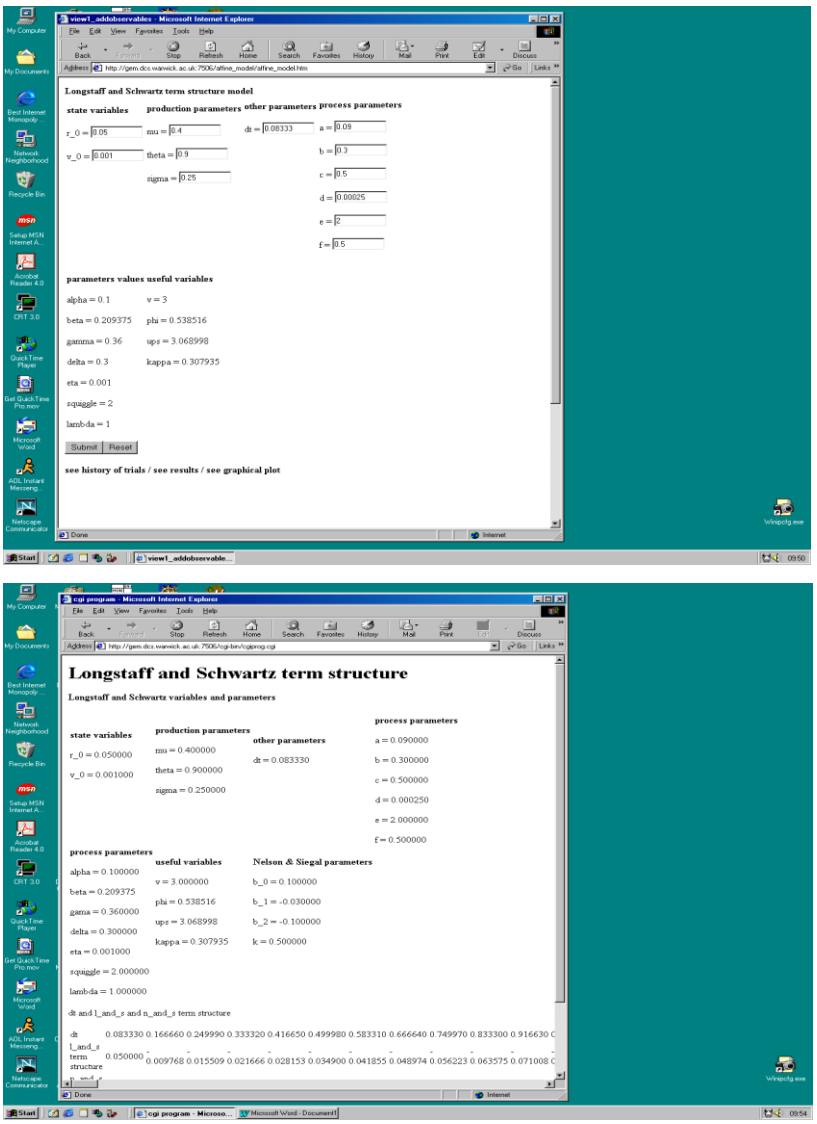


Figure 7.5 The web-based model

7.2.2 The dtkeden model

The dtkeden model was developed in an attempt to overcome the limitations of the spreadsheet and web based models described above. The qualities of this model include: i. support for interpersonal communication; ii. the de-centralisation of the modelling activity; iii. computer support for playing the roles of agents; iv. means for managing the history of model development and use (management of different trials of parameters); v. scope for enhanced visualization and exploration of output results.

i. *support for interpersonal communication:* The use of the model in a classroom or by practitioners demands support for various forms of interpersonal communication depending on the context and uses of the financial model. Interpersonal interaction is very important in the development of software systems including systems supporting the financial engineering[8] activity. This is widely recognized in [Sal87, Bos89, VF87]. The development of systems meeting various objectives and purposes relies on group social activity that involves various roles and modes of interaction among participants [AC98, Flo87, XIA98]. The research of Sonnenwald (1993) on 'communication in design' reveals that many design situations involve a group of participants interacting with each other in developing an evolving prototype of a system. Sun (1999) proposed dtkeden as a collaborative tool that facilitates interpersonal interaction and knowledge creation amongst participants. However, he stressed that computer-based interpersonal interaction should not be regarded as a replacement for conventional means of communication but as an important enhancement of existing ones.

ii. *Decentralisation of the modelling activity:* In a single user environment, the modelling activity is centred around one person. In order to derive sensible meaning from the model, the modeller needs to be able to play various roles and for this many different kinds of specialist expertise are required. For example, in the practitioner's scenario described in section 7.1.4 (cf. Figure 7.4), in the course of engineering a new financial instrument tailored to the need of a client, the modeller has to play the role of the client, the salesperson the risk analysis manager, the quantitative analyst, and the trading expert. Similarly, in the classroom scenario the modeller has to play the role of the teacher and student interchangeably. This creates a heavy load on the

[8] In this chapter, the terms 'financial engineering' and 'financial instruments modelling' are used interchangeably. Financial engineering is considered as the application of engineering principles in modelling financial instruments.

modeller and may result in an individual bias in the modelling activity. This individual bias is considered in Gruber and Sehl's shadow box experiment described in [Goo90]. This motivates the decentralisation of the modelling activity to alleviate the load on the modeller and to redress the individual bias.

iii. *Computer support for playing the roles of agents:* There is a benefit in the modeller being able to appreciate the roles of all agents participating in the financial engineering activity at first-hand. The dtkeden model can be constructed in such a way that each client can be configured to reflect the working environment of a participant (cf. the railway model). The modeller can also act in the role of a super-agent sited at the server and capable of gaining insight into the actions of participants through simulating changes to the environment, setting up scenarios, changing agent privileges, and monitoring and resolving conflicts in interaction as and when arising. Agency control is targeted with privileges for access and agency action set up by an external observer (the super-agent at the server) and enacted by internal actors (participants at the clients). In DEM, participants partake in the modelling activity by playing their own role and by collaboratively resolving conflicts in their shared understanding.

iv. *Management of the modelling activity*: Management and documentation of the modelling activity are an important consideration for tracking the history of the model development and directing its future evolution. Management of the modelling activity can take various forms. For example, managing a paper based modelling activity involves arranging the piles of papers describing the model according to various criteria. In computer-based modelling, the management activity is more elaborate and takes the form of storage, retrieval, saving, and documentation. The management of a model becomes more and more important with time-lapse and model evolution. In dtkeden, shared management of the modelling activity is possible through the communication of definitive scripts that involve file manipulation.

v. *Visual exploration of the resulting output*: tabulated results and 2D visualization give little support for the exploration and the identification of hidden pattern of relationship between data. Today, advanced visualisation techniques including the use of self organized maps [DK98], attribute explorer[9], and virtual reality [Car00, Car99], promise a richer medium to convey properties of the input and output data and the

[9] Originally developed by IBM, UK due to an idea of Bob Spence, Imperial College, London. The Attribute Explorer is implemented using EM tools by Roe (2000).

pattern of their relationship as the parameters of the model varies. Despite the lack of sophistication in visual effects in the dtkeden tool, a rich potential for interaction with the visual representation of observables in distributed views is possible (cf. The EM Attributed Explorer).

The following paragraphs describe the design and implementation of the stand alone affine interest rate model and its distributed variant for future use in a classroom context. Although the features of the distributed model are not limited to the interface but can be extended in an open ended way by sharing definitive scripts across the network, the interface design is essential to convey the features of the model. The interface is split into three parts the stand-alone, the toolbox, and the distributed part. The standalone part is the same for both the teacher and the student. It includes the parameters of the model and the graph plot. The toolbox is also the same for the teacher and student and it includes the option of updating the parameters of the model, saving the current trial of parameters, loading previous trials of parameters, and printing out the tabulated results of the fitted model. The distributed part differs for the teacher and student. The difference arises from the various roles attributed to the teacher and the student. The teacher, at the server, plays the role of the external observer in the DEM who sets the context of interaction of various internal agents (the students). The context of interaction is created by setting privileges for access assigned to various students, located each at a client workstation. The teacher can i. select a student, ii. enable / disable server propagation, iii. add / substract handles and iv. send messages to this student:

i. The teacher can select a student by typing his / her name in the `user` or `send to` input text boxes.

ii. Enable / disabling server propagation is used in the client-server architecture of dtkeden to enable / disable the propagation from the server to the client of any change made to observables at the server. By default, any change to the model introduced by the teacher is propagated to all students. If the teacher wishes to work on his/her own then there should be a mechanism that stops all propagation to the students. This involve Enabling / disabling server propagation.

iii. Adding / subtracting handles involves giving permission to a student to change or see the value of an observable (observables can be parameters in the Longstaff and Schwartz or Nelson and Siegal models).

iv. Sending messages to the students can take place by typing the name of the student in the `send to` text box and including the definition / statement to send in the send message area. Similarly, the teacher can receive message from students.

The student can select another student to send to / receive from messages. He / she can also send / receive messages from the teacher.

The following snapshot shows the interface of the teacher view in the distributed model.

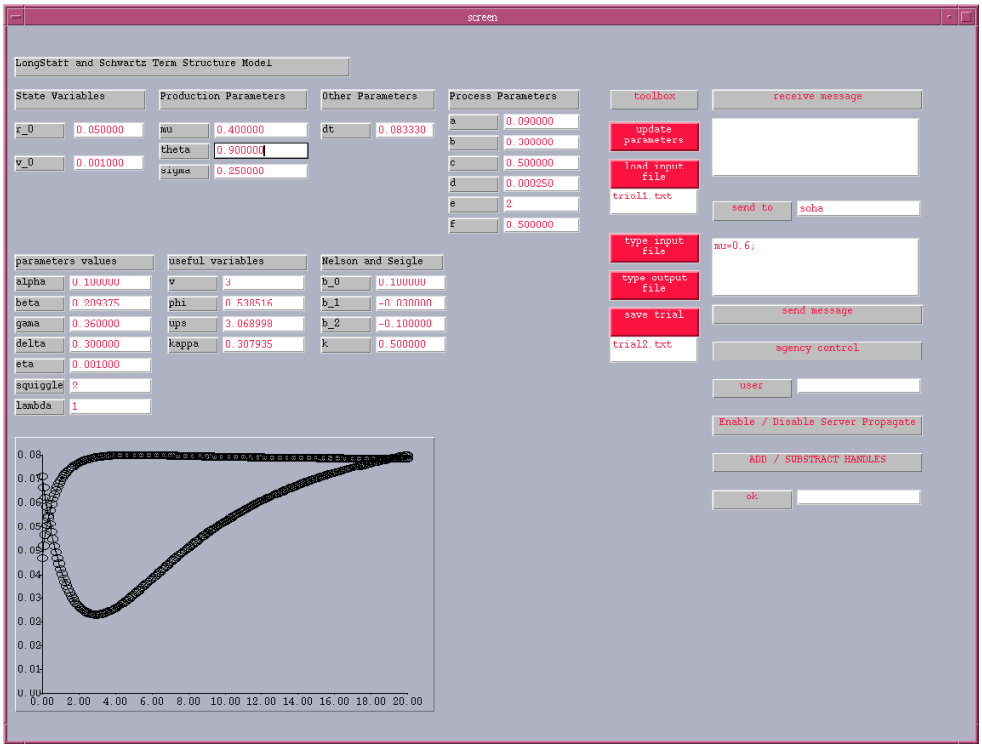


Figure 7.6 A snapshot of the teacher view in the distributed model

The interface of the teacher and student differs in the communication part as follows.

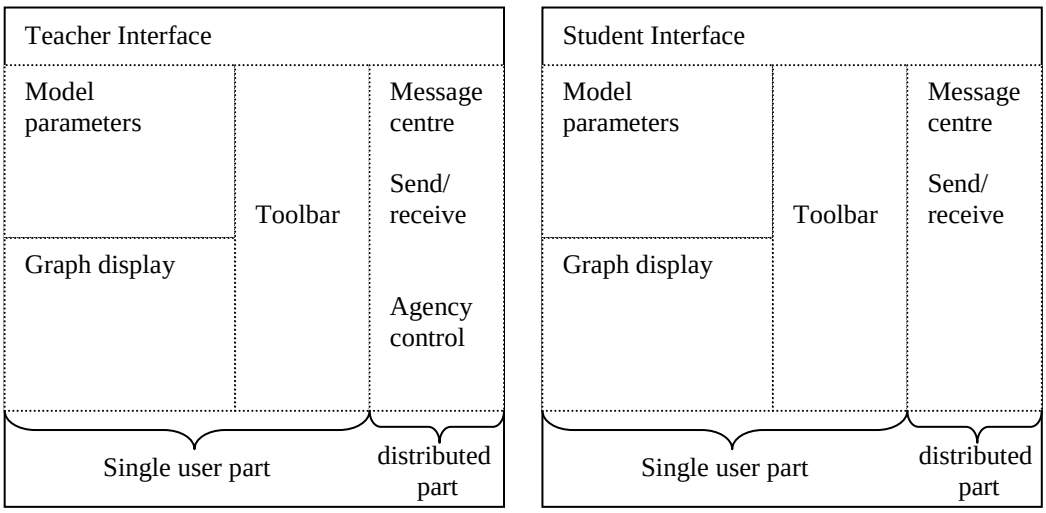


Figure 7.7 Views of participants in the distributed model

The development of the tkeden single user model is very straightforward and takes a short time once the developer has some familiarity with the tool. The interaction with the model is not limited to the visual interface. The model can enter new definitions for parameters, alter existing ones, and assign new values to parameters. This is facilitated by the tkeden window.

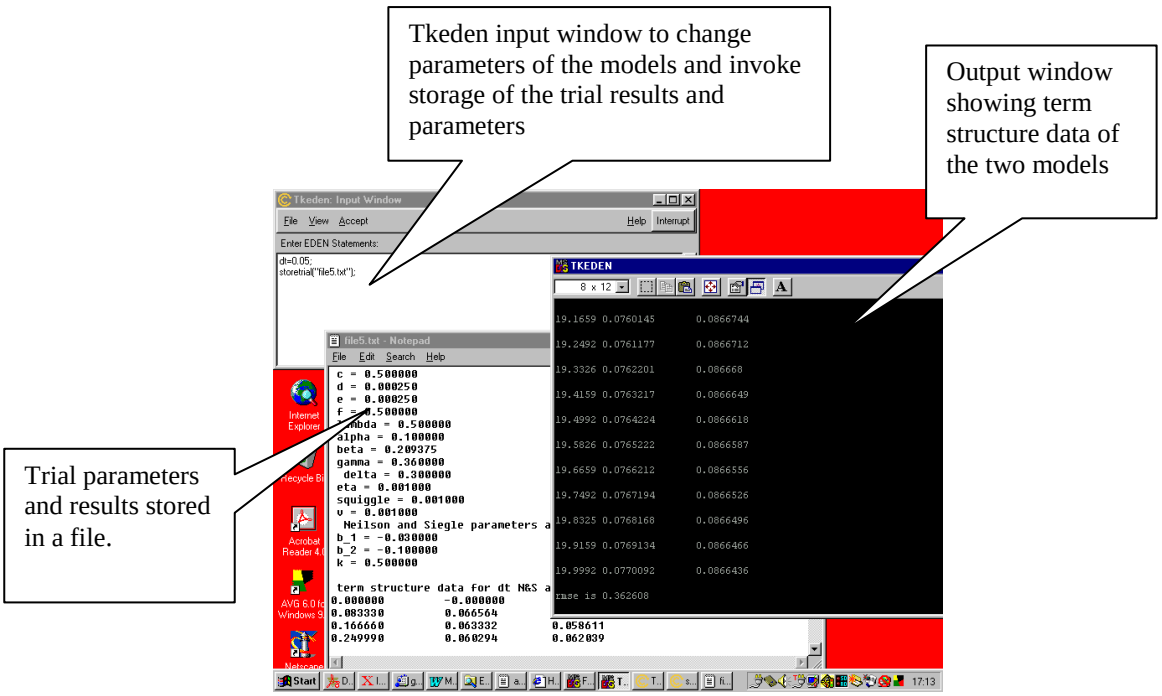


Figure 7.8 Open ended interaction with the model beyond the visual interface

Any parameter of the Longstaff and Schwartz or the Nelson and Siegal model can be changed and consequently their term structure data is changed. This is enabled through an event triggered action (cf. table below) that updates term structure data following a change in a parameter model value.

```
proc update:r_0,v_0, mu, theta, sigma, dt, a,b,c,d,e,f,lambda,b_0,
b_1, b_2, k {
 for (i=1;i<242;i++) {
   if (i==1){
   A_t[i]=0;
   B_t[i]=0;
   C_t[i]=0;
   D_t[i]=0;
   hphi[i]=0;
   hups[i]=0;
   taw[I]=0;
   nands_rate[i]=b_0+b_1;
   lands_rate[i]=r_0;
   Writeln(taw[i],"\t",nands_rate[I],"\t",lands_rate[i],"\n"); }
  Else
  {taw[I]=taw[i-1]+dt;
```

```
 nands_rate[i]=b_0+(b_1+b_2/k)*((1-exp((-1)*k*taw[i]))/(k*taw[i]))
               -(b_2/k)*exp((-1)*k*taw[i]);
 hphi[i]=2*phi/(exp(phi*taw[i])-1);
 hups[i]=2*ups/(exp(ups*taw[i])-1);
 A_t[i]=1/(((delta+phi)/hphi[i])+1);
 B_t[i]=1/(((v+ups)/hups[i])+1);
 C_t[i]=((2*alpha/(v+ups+hups[i]))-(2*beta/(delta+phi+hphi[i])))
        /(beta-alpha);
 D_t[i]=-2*((1/(v+ups+hups[i]))-(1/(delta+phi+hphi[i])))
        /(beta-alpha);
 lands_rate[i]=-(2*gamma*log(A_t[i])+2*eta*log(B_t[i])
               + kappa*taw[i]+C_t[i]*r_0+D_t[I]*v_0)/taw[i];
 writeln(taw[i],"\t",nands_rate[i],"\t",lands_rate[i],"\n");}
 rmse=rmse+(lands_rate[i]-nands_rate[i])
      *(lands_rate[i]-nands_rate[i]);
  }/*end for*/
 writeln("rmse is ",rmse);
}
```

The term structure data for the Longstaff and Schwartz and the Nelson and Siegal model is stored in a file. Results of different trials can be saved in separate files.

The multi-user model can be automatically established from the single user model by starting a server and including the teacher script at this server. Several clients can be connected to the server (within network capacity) each including the student script.

```
For (i=1;i<=CLIENT_LIST#;i++)
{sendClient(CLIENT_LIST[i],"include(\"studentscript\");");}
```

In the above script, CLIENT_LIST refers to the list of all connected clients (students) to the server (teacher). Modellers at different client workstations can change the parameters of the model in their own view and share their results by communicating their experience in interaction with the model. This communication is enabled through the send/receive feature in the message centre in the visual interface. The communication between the modellers at each client workstation is open ended. There is no pre-conceived workflow for the communication between different modellers. This makes the distributed version of the model open for use at a classroom or institutional level. The distributed model could be used to identify structured patterns of communication between modellers. Once this pattern can be repeated reliably, a more circumscribed model can emerge fulfilling a particular objective: educational, or decision support for financial product development.

The teacher (at the server) creates the context of interaction between students by adding / subtracting handles from each student and by enabling or disabling server propagation.

```
 /* to enable / disable the propagation of any change to the model at
      the server from reaching the student model, the variable
                 propagateType is set to 1 / -1 */
propagateType=-1;
  /* The context of interaction is set by the teacher at the server
   through the visual interface, or by direct script inclusion */
propagateType=1;
addAgency("soha", "oracle" , "mu");
addAgency("soha", "handle" , "mu");
 /* this will add agency to the client soha. This agency enables the student, soha, at the client
  to see and manipulate the parameter mu.  this agency could be also introduced at the server
                                using the lsd notation */
%lsd
     agent soha
     oracle mu
     handle mu
```

Figure 7.9 depicts the client-server configuration of the distributed model. Modellers can share experiential knowledge about the interaction with the model by passing possible parameter changes to each other or by exchanging files storing parameters and results of previous trials. Privileges to change the parameters of the model can be assigned to each modeller by declaring shared observables as being oracles or handles to a modeller.

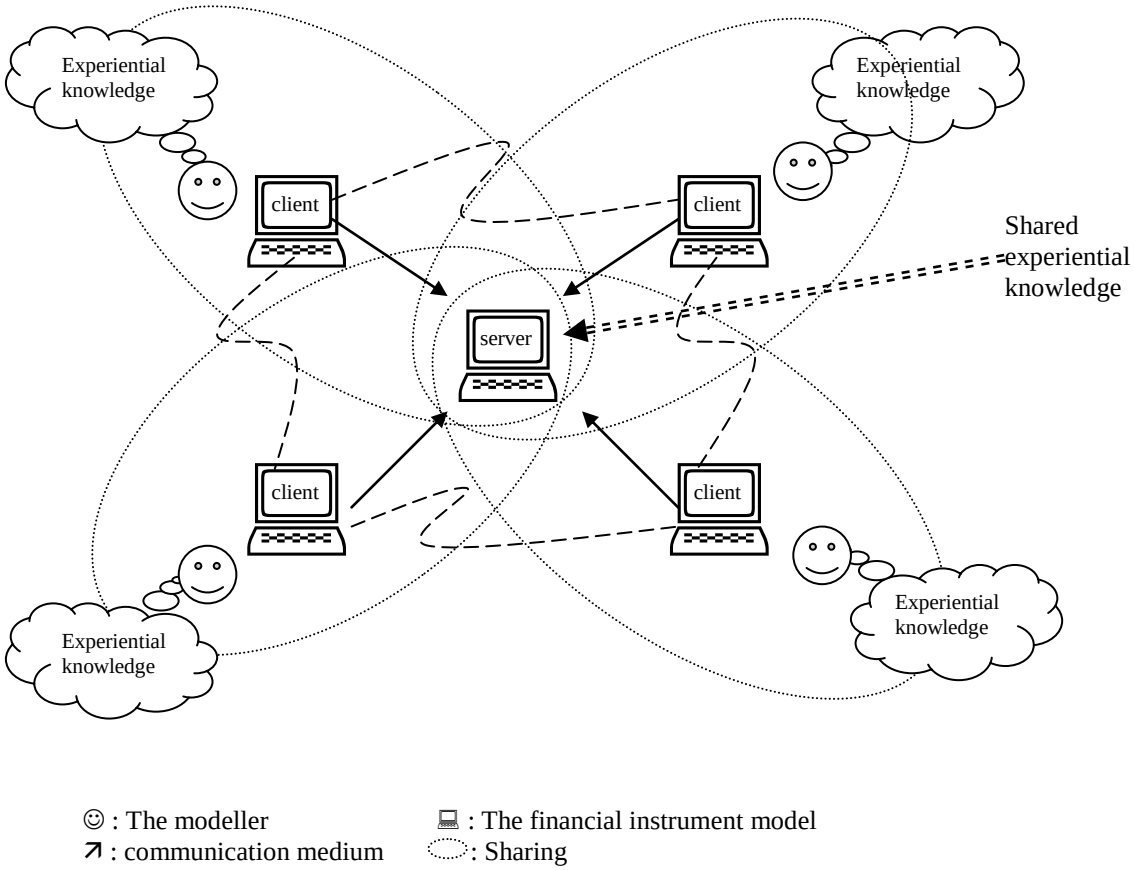


Figure 7.9 The distributed model for sharing experience

A possible communication between modellers might take the following form:

```
sendClient("soha"," dt=0.05; savetrial(\"file1.txt\");");
sendClient("meurig"," lambda=2;");
sendClient("jaratsri"," r_0=0.1;");
```

7.3 Distributed EM for Modelling Financial Instruments

Chapter 3 briefly introduced Distributed EM (DEM) enacted through the distributed EM tool dtkeden. DEM contributed to the motivation of a paradigm shift in Software System Development (SSD) and broad implications of EM in meeting the wider agenda of computing. The railway accident model, the virtual electrical laboratory (VEL), the warehouse model are case studies that reveal the potentially rich contribution of DEM to studies in computer mediated interpersonal interaction, computer-based group learning, and computer supported participative business process modelling. In these case studies, modellers are internal agents shaping the state of the model by interacting from within their own views and within their attributed agency created by an external modeller having a global view and superior agency power. Adopting a client server architecture, the internal modellers are at the clients and the external modeller is at the server.

This section discusses the claim that:

> "***DEM has important features that motivate broad implications on principles and foundations for establishing virtual environments[10] for financial engineering[11]. This is supported by examining the wider needs of the financial engineering activity considered in its broader social context that involves a network of collaborating financial experts with various roles and privileges of action . The claim will be supported by referring to the role of DEM in the***

[10] The term 'environment', used above, designates a medium grouping a set of features and visited to achieve a particular objective. The attribute 'virtual' is used to inform that this environment is not a real physical environment but a computer-based one. This does not eliminate the correspondence between the computer-based environment and its real world referent. In the EM literature, such types of environments are often referred to as 'computer artefacts'.

affine interest model case study and various financial instruments modelling practices"

Distributing the modelling activity help in overcoming the limitations of centralized modelling, provides greater support for interpersonal communication and group learning and decision support, and redresses the individual modeller bias.

Distributed modelling relies on an established framework and a supporting technology for enacting this framework. Such a framework could be based on: i. a social science theory; ii. a view on the mode of interaction and role of different participants in multiagent systems; iii. the type of relationship between modellers; and iv. an appropriate definition of the role of agency in the modelling activity. The supporting technology for enacting the framework of distributed modelling relies on an appropriate network communication supporting different modes of interaction.

i. The DEM framework [Sun99] is based on the theories of distributed cognition[12] [Hut95] and ethnomethodology[13] [Gar67].
ii. Modellers collaborate as internal agents shaping the state of the model through views and privileges of actions. These privileges are set by an external modeller that creates the context for interaction.
iii. DEM supports a collaborative relationship between modellers, where individual modellers share insight in an open ended environment for collaboration. This environment is not constrained by a rigid mode of preconceived interaction.
iv. Agency gives modellers privileges to view observables in the model (oracle) and exert agency actions that introduce state change to observables in the model (handles).

The DEM framework is enacted using the dtkeden tool that features:

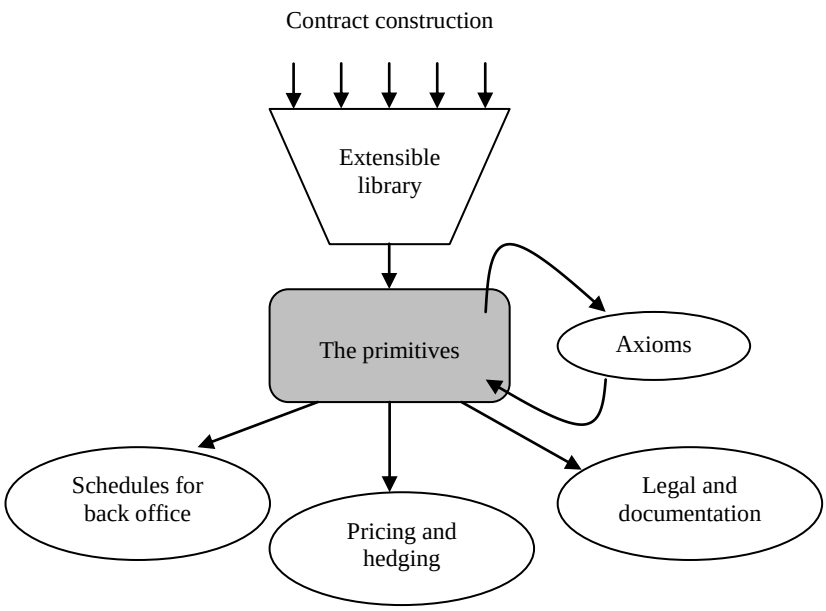
- Distributed communication of definitive scripts
- The support of various modes of interaction (the broadcast mode[14], the private mode[15], the interference mode[16], and the normal mode[17])

[11] This would refer to the objective of distributed modelling of financial instruments.
[12] The concept of distributed cognition represents a synthesis of cognitive, anthropological and social scientific approaches to the study of collaborative works. Its central theme involves locating cognitive activity in context, where context is not a fixed set of surrounding conditions but a wider dynamical process of which the cognition of an individual is only a part.
[13] The empirical investigation ('ology') of the methods ('method-') people (ethno) use to make sense of and at the same time accomplish communication, decision making, reasonableness, and acctions in every day life.

- Client server architecture for network communication

As discussed in chapter 4, DEM provides principles that support situated context dependent computer mediated collaboration. In the finance context, collaboration in modelling financial instruments is rich in interaction and features specific context dependent characteristics, patterns of interaction, and modes for sharing insight. This is best motivated by considering the rich level of interaction invoked in modelling financial instruments using the computer-based language developed by Jones et al (2000):

> *"a formalized language to describe financial contracts*[18] *is being developed with the aim of facilitating back office contract execution, accelerating in a substantial way risk analysis, graphically representing a contract as a decision tree, reminding the front-office about any potential exercise decision that has to be taken, and generating intuitive simulations for the marketing departments of investment banks". Jones et al (2000)*



[14] This is the most primitive style of interaction between modellers [Sun99]. Messages are broadcast from the external modeller, at the server, to all internal modellers, at the client. This style of interaction resembles electronic group meeting without video or audio.
[15] Supports a one-to-one interaction between the external agent at the server and an internal agent at a client [Sun99]. One-to-one interaction plays an important role for managing different perspectives in a group [Son93].
[16] The interference mode [Sun99] concerns real world situations where many-to-many interactions are the norm and singular conditions require the intervention of a God-like superagent.
[17] Interpersonal interaction is mediated by the computer with reference to specified privileges of modellers to access observables.
[18] In the finance literature the terms 'financial instruments' and 'financial contracts' are used interchangeably.

Figure 7.10 Features of the formalized language to describe financial contracts proposed by Jones et al (2000)

Figure 7.10, reproduced from Jones et al (2000), illustrates the main features of the formalized language to describe financial contracts. The figure inspires a complex workflow of the financial instrument model between various departments in the financial institution (back office, front office, legal department, and pricing and hedging department).

This motivates further research and in the area of: i. computer-based interpersonal communication for modelling financial instruments; ii. modes of interaction in group financial engineering activity; iii. the role of agency in the distributed financial modelling activity; and (iv) computer-based group decision support for the financial instrument modelling activity.

Sonnenwald has raised many issues related to interpersonal interaction in design [Son93, Son99] that are of particular relevance to the DEM framework:

- The need for diverse modes of interaction between group members participating in a design process in order to develop a comprehensive understanding of the design and facilitate multiple exploration of knowledge.
- The need for diverse roles and interaction networks for inter-group and intra-group members and diverse styles of interaction between modellers in each stage of the design.
- The escalating complexity of the architecture of interaction among modellers from multiple disciplines, domains and backgrounds participating in large systems modelling and design.

The above issues raised by Sonnenwald reveal the challenge facing traditional computer-based support for interpersonal interaction that resorts to pre-conceived mode of interactions enabled through rigid interface design.

DEM considers the distributed modelling activity in its wider social context, where situated interpersonal interaction can be established at any instant through the communication of definitive scripts. The context of interaction is created through agency privileges set by an external modeller. Despite the limited modes of interaction supported by DEM, growing needs for distributed modelling can be cultivated through continuous interaction among modellers to identify a reliable pattern of interaction that may inform at a later stage traditional structured approaches to computer-based support of interpersonal interaction.

By referring to the simple case study of modelling the interest rate using the affine framework, considered in a classroom scenario, different lessons can be learned:

- An external modeller is needed to create the context of interaction. This could be the teacher in a classroom scenario or the general manager in the financial enterprise

scenario. This external modeller sets the privileges for interaction of various modellers in a situated way.

- Styles and modes of interaction among a group of participants are difficult to frame in prescribed actions. Various needs of interpersonal interaction can be supported within a flexible distributed modelling framework. This framework can accommodate various modes of interaction as and when they arise.
- Distributing the modelling activity serves a better educational, operational, and decision making objective in the finance context.

7.4 Summary and Outlook

This chapter considers the case study of an affine term structure spreadsheet model originally developed by Nick Webber. Two models are developed using web-based and distributed EM Technology.

Empirical Modelling technology addresses basic issues related to the requirements engineering of computer-based environments for modelling of financial instruments. Its basic contributions are in laying principles and foundations for a distributed framework for modelling financial instruments and in presenting a proof of concept technology for enacting this distributed framework. Such an environment would better support the financial instrument modelling activity and widen the contribution of the computing activity beyond the implementation of the mathematics of the financial model to providing group learning, group decision support and knowledge construction.

Chapter 8

Empirical Modelling Technology And Data Intensive Financial Applications

8.0 Overview

This chapter discusses the prospects for using Empirical Modelling Technology for data intensive financial applications. The context for this discussion is established in section 8.1, which considers a financial case study on tests for market efficiency. The data collection and financial analysis for this case study was conducted by Keng Yu Ho, while the technical programming work was undertaken by the author. Section 8.2 discusses the broad implication of EM technology on the finance research development cycle. Section 8.3 discusses the prospects towards applying EM technology in connection with data intensive financial applications.

The chapter concludes with the research agenda for using Empirical Modelling technology as a framework for computer-based support of data intensive financial applications.

8.1 The Case Study

8.1.1 Motivation and objective

The computer implementation of different techniques and methodologies used in financial research is rarely documented in the research itself or separately. This can be mainly attributed to the importance of result reporting and analysis in finance, which takes precedence over the computer implementation. Financial experts are not much concerned to reveal their adopted technology support as this will not give any value added to their results and inferences. AI technology is currently a pioneer technology in bridging the gap between financial analysis and computer support to this analysis. Implementation of algorithms, performance, and speed come first in the agenda for use of computers in finance. This gap in the literature on computer-based support in finance has serious drawbacks:

(1) The lack of adequate and complete reporting of computer-based support to financial analysis and research limits the scope of the computer-based support to the financial research activity and hides important aspects of the relationship between the finance research development cycle and its corresponding software system development.

(2) Limiting the computer-based support for financial research to the implementation of algorithms and econometric methodologies using fourth or third generation languages (e.g. spreadsheet and statistical packages, or programming using C or Fortran) diverts attention from providing semi-automated support for financial research and analysis. Providing such semi-automated support relies heavily on the coherent integration of the human and automated activity that makes it possible to infer meaning and conclude results in the course of financial analysis.

(3) The reliance on structured methodological approaches in financial analysis distracts attention from providing computer-based support for experiential knowledge construction that is guided by human insight and subjective understanding.

(4) The lack of concern for appropriate reporting of the computer-based support for the financial research obscures the links between the computing and finance domains and puts disproportionate emphasis on computer-based implementation of algorithms. This inhibits emerging technologies from offering wider support for financial research activity that is more intimately related to the computer-based activity and in particular to context dependent software system development.

This chapter considers a case study that is typical in data intensive financial analysis and computer-based implementation of various methodologies and algorithms. The case study serves three objectives that contribute to the theme of the thesis:

(1) To uncover quantitative and qualitative aspects of computer-based support for the financial research development cycle. Quantitative aspects refer to metrics on the amount of data processing and file manipulation as well as to issues related to the technical implementation of algorithms and methodologies. Qualitative aspects refer to the integration of the human and automated activity throughout the whole of the finance research development life cycle.

(2) To identify the prospects of EM technology in supporting the finance research development cycle and in particular in establishing a closer integration between the finance research development and software system development activity.

(3) To inform the development of the future generation of EM tools in respect of requirements for data intensive financial analysis.

8.1.2 About the case study

The research work described in this section was jointly conducted with Keng-Yu Ho[1]. All the financial analysis and data collection was undertaken by Keng-Yu Ho, whilst the computer implementation of econometric models, web development, and the electronic manipulation of relevant data sets, was carried out by the author. The outcome and progress of the research is framed within a web-based environment (cf. [Maa02-Web]) serving a documentary purpose.

As introduced in chapter 2, market efficiency refers to the extent to which security prices fully and correctly reflect all relevant information [Mal99]. This implies that it is impossible to make abnormal profits on the basis of that information set. All the empirical research on the theory of efficient markets has been concerned with whether prices fully reflect particular subsets of available information. Tests for weak form market efficiency were conducted with past price histories as the information subset of interest [Fam70].

Research on testing market efficiency is broad and encompasses testing the profitability of certain trading rules, testing asset pricing models, adopting event studies or examining stock market anomalies. The research considered in this thesis is confined to long-horizon event studies[2]. Event studies, introduced in Fama, Fisher, Jensen and Roll (1970), provide evidence on how corporate events affect stock prices [Ho00]. Two types of event studies can be

[1] Doctoral researcher, supervised by Dr. Abhay Abhyankar at Warwick Business School, in UK.
[2] Event studies are conducted to test the semi-strong form of market efficiency.

conducted: short-horizon (known as standard event studies) or long-horizon. Standard event studies assume that the response of prices to an event is short-lived, whereas long-horizon event studies focus on the impact of corporate events (such as rights issues, convertible debt issues, mergers, etc.) over an event window of up to 5 years. Further details on the historical overview and evolution of long-horizon event studies can be found in [Ho00].

The following paragraphs describe the financial research development cycle (FRDC), focusing on particular stages of the cycle (cf. Figure 8.1). This description highlights essential aspects of the FRDC such as:

- The size and amount of data processing (this aims at informing the requirements for data intensive computer-based applications)
- The programming paradigms used
- The methodologies adopted

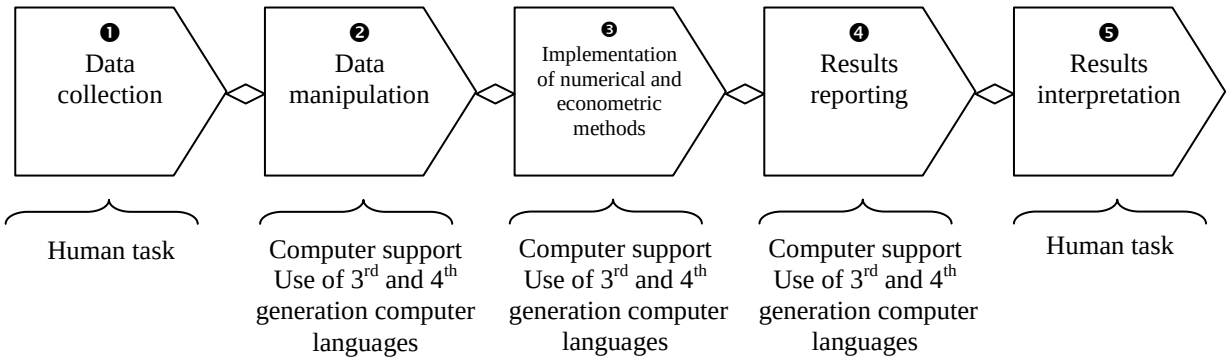


Figure 8.1 The Research Development Cycle

❶ The data collection / manipulation stage:

The research work undertaken by Ho focuses on long-horizon event studies conducted using data from the UK market. The rights issues event (RI) is considered. UK data on rights issues was taken from Extel Takeovers, Offers and New Issues. *Returns* and *market capitalization* data is taken from the London Share Price Database (LSPD)[3] and *Book-to-market ratio* data is taken from Worldscope[4]. Details on the legal and procedural aspects related to rights issues event are provided in [Ho00, AH01].

In conducting the rights issues event study, collected data was stored in three types of file: **`event firm`** data files (including all firms having a rights issue event in a particular year and month over the period of study); **`book-to-market ratio`** data files; and

[3] http://www.lbs.ac.uk
[4] http://www.primark.com

`excluded firms` files (event firms excluded from the matching approach[5]). All files are saved as text and their data corresponds to a particular year over the period (89-99).

The amount of data stored in the files used is of the order of 100MB. The size of the returns file used (RETS1999 from LSPD) is 45MB. It consists of 6911 firms having data from 1955 till 1999. This file holds approximately 16KB of data for every firm.

The **`event firm`** files are relatively small, and include a little information about the event firm (its unique number in LSPD, and the date of the event). Each year, from 1955 till 1999, has its own **`event firm`** file. The maximum number of firms in an **`event firm`** file is around 95 firms.

`Book-To-Market ratio` (BTM) files are also relatively small. They include only BTM data and sedol numbers (unique identifier number of a firm in LSPD) for a maximum of 2000 firms. BTM files are also constructed annually.

`Excluded firms` files contain only the sedol number of a maximum of 400 excluded firms.

The manipulation of the returns file of the LSPD entails filtering selected fields, searching for specific firms, and extracting a specific range of returns.

❷ The implementation of different techniques and methodologies:

Conducting long horizon event studies involves the estimation of long horizon abnormal performance of stocks following or prior to the event date. Different methodologies [AH01] can be used to assess this abnormal performance. These include:

i. *buy-and-hold abnormal return* (BHAR)[6], which computes the difference between the compounding return of the event firm and that of the benchmark over a period of time (3-5 years after the event);

ii. *cumulative abnormal return*[7] (CAR), that simply adds up the returns of the event firm in its benchmark and computes their difference as traditional short-horizon event studies.

---

[5] Firms that had a similar event in the previous 3 years before the event firm year are excluded from the matching approach.

[6] $BHAR_i = \prod_{t=1}^{T}(1+R_{i,t}) - \prod_{t=1}^{T}(1+R_{benchmark,t})$ where $R_{i,t}$ is the return for sample firms.

The mean buy-and-hold abnormal return is the weighted average of the individual *BHAR*s:

$$\overline{BHAR} = \sum_{i=1}^{N} w_i \cdot BHAR_i$$ where w_i is the weight (either equal-weight or value-weight).

iii. *calendar time abnormal return* which measures abnormal returns relative to the expected returns from a chosen asset pricing model (the Fama-French three factors model[8]) .

Conventional t-statistic[9] is used, and p-values, based on skewness-adjusted t-statistics are recorded.

In assessing the abnormal performance, using CAR and BHAR, three types of benchmark firms are used:

(1) The matched size / industry benchmark: the benchmark is taken as a matching firm in the same industry and having a closest but higher market capitalization than the event firm in the corresponding event year.

(2) The matched book-to-market ratio / industry benchmark: the benchmark is taken as a matching firm in the same industry and having a closest book-to-market ratio to the event firm in the corresponding event year.

(3) The matched portfolio benchmark: the event firm is matched with one of 25 size / book-to-market portfolios. These portfolios are obtained by ranking all firms from LSPD for which market capitalization data is available for the end of the previous calendar year and for which book-to-market ratio data is available at the end of the previous fiscal year. Firms are allocated into a 5x5 grid and the 25 value-weighted

---

[7] $CAR_{i\tau} = \sum_{t=s}^{s+\tau}[R_{it} - R_{benchmark}]$ where R_{it} is the simple monthly return for sample firm *i*, $R_{benchmark}$ is the simple monthly return on the benchmark that corresponds to the firm i.

[8] For each calendar month, form the portfolio that contains all the event firm returns within the 3-5 years period of study, the FF 3-factor model below is used to estimate the abnormal return:

$$R_{pt} - R_{ft} = \alpha_i + \beta_i(R_{mt} - R_{ft}) + s_i SMB_t + h_i HML_t + \xi_{it}$$

R_{pt} is the monthly return on equal-weighted or value-weighted claendar time portfolio
R_{ft} is the monthly return on one month treasury bills
R_{mt} is the return on the return of the value-weighted portfolio of all firms in the market
SMB_t is the difference in the returns of the value-weighteed portfolios of small stocks and big stocks
HML_t is the difference in the returns of the value-weighted portfolios of high book-to-market stocks and low book-to-market stocks
α_i, β_i, s_i, h_i, and ξ_i are parameters of the regression.

[9] $$t = \frac{\overline{BHAR}}{\sigma(BHAR)/\sqrt{n}}$$

where $\overline{BHAR}$ is the mean and $\sigma(BHAR_\tau)$ is the cross-sectional standard deviation of abnormal returns for the sample of *n* firms.

portfolio returns are calculated for each month. The event firm is then matched to a portfolio with similar size/book-to-market characteristics.

The following paragraphs elaborate on the programming paradigms used in assessing the abnormal performance of event firms computed using different methodologies and choice of benchmarks. General features of the developed programs are presented and some software metrics are provided:

Third and fourth generation languages (C and Excel) are used in the computer implementation of the different methodologies and benchmarks used in assessing the abnormal performance of event firms. The programming paradigm used in the case study is the imperative[10] one, and the C programming language is used. The C programming language is chosen for speed, ease of use, and compatibility with available financial libraries. The spreadsheet application Excel was used for testing the correctness of generated outputs and for conducting statistical analysis on output results.

All programming languages have their good and bad points, and consequently the choice of a particular language for the implementation of econometric and numerical methodologies has advantages and disadvantages. Modern programming languages have many similarities in types, declarations, expressions, statements and data structures. By using relatively few language constructs and avoiding implementation decisions based on peculiarities of C, we can develop programs that can be easily translatable to other languages. The advantage of C is that it is widely used and has all the basic features needed in various implementations. C has a rigorous high level syntax that allows easy identification of the main features of the program [Sed90].

The imperative programming paradigm was adequate to the program development done so far. The designer was very careful in specifying the program requirements for implementing the econometric and numerical methodologies adopted in terms of input – process – output. This makes it appropriate to use an imperative programming language. Data management was a great concern. The UK dataset was provided in textual format – this made the file processing approach the most straightforward to use.

The choice of a proper data structure is a major decision in the implementation stage. Declaring and using abstract user defined data types was constrained by memory capacity.

[10] An imperative language uses a sequence of commands to carry out the desired operations.

At an initial stage of program development, object oriented programming (in JAVA or C++) and design would not have had a great impact on the analysis and programming stages. Declaring a user defined abstract data type for a firm was quite sufficient.

Arrays (one dimensional and multi-dimensional arrays) and user-defined types (implemented using the data structure **`struct`** in C) were the main data structures used. Examples of user defined data types used data on firms read from different files (LSPD RETS, constructed Book to Market ratio (BTM), and event firms files) are provided in the following table.

Table 8.1 User defined data types used in methods implementation

User defined data type for LSPD RETS firm	***User Defined data type for Event firm***	***User Defined type for BTM firm***

`typedef struct`
`{int ncomp;`
`int sedol;`
`char rname[33];`
`int dti;`
`int freq;`
`int start;`
`int end;`
`int nobs;`
`int samp;`
`int indy;`
`int mval;`
`int ntrade;`
`int ftpr;`
`int dyld;`
`int per;`
`int dmark;`
`int netass;`
`int spread;`
`float zret[540];`
`int dates[540];`
`int mcap[45];`
`} lse_companies;`
	`typedef struct company`
	`{int starting_index;`
	`int event_year;`
	`int event_month;`
	`int sedolnum;`
	`} company ;`
		`typedef struct`
		`{`
		`int sedol_number;`
		`char btm_name[33];`
		`int btm_mcap;`
		`int btm_startyear;`
		`int btm_endyear;`
		`float btm;`
		`int btm_indy;`
		`char marker[2];`
		`} btmdata;`

The *lse-companies* user defined data type represent a company in the LSPD returns file RETS. It has 18 fields, and 540 returns, dates, and market capitalization taken from year 1955 until 1999. The user defined data type *company* represents an event firm. The event year, month, and the sedol[11] number identify the event firm. The user defined data type *btmdata* represents data for a firm in the book-to-market ratio file.

*The standard C libraries (*stdio.h, stdlib.h, ctype.h, string.h, float.h) were used and some user defined mathematical and selection functions were implemented.

The file data management paradigm was adopted and large LSPD text files were read and saved as *binary files* for ease of access and search.

Basic building blocks of programs implementing the different methodologies to compute the abnormal performance using a particular choice of benchmark are:

- *matching:* used when taking the benchmark as book-to-market / industry match or size / industry match
- *sorting* : used when taking a 25 size / book-to-market portfolios benchmark and when adopting an asset pricing model to assess the abnormal performance of event firms
- *bootstrapping :* used to compute the skewness-adjusted t-statistics
- *filtering* : used in the matching and for excluding certain firms

The programs developed are:

(1) Programs to find different benchmarks for an event firm:
   - a matching firm with nearest book-to-market ratio and same industry group
   - a matching firm with nearest book-to-market ratio and same industry group
   - a portfolio with same similar size and book to market characteristics

(2) Program to find compute adjusted t-statistics for computed abnormal returns:

(3) Program to find the portfolios used in the asset pricing model (Fama French 3 factors) to compute the abnormal performance of the event firm

A brief description of some of these programs is given below, specifying each program in terms of input-process-output. In the following description MCAP refers to market capitalization, BTM to book to market ratio, Sedol to a unique 8 digit number identifying a UK firm in LSPD, and RETS to LSPD returns file.

Size / industry matching: This program takes as input event year data (event date, firm sedol), excluded firms data (sedol), returns data (LSPD RETS), and produces an output file showing for every event firm up to 10 matching firms (5 up, 5 down) in the same industry group. The matching between firms is based on market capitalization information.

Book to Market / industry matching: This program takes as input event year data (event date, firm sedol), excluded firms data (sedol), returns data (LSPD RETS), BTM

[11] Every firm in LSPD has a unique sedol number that identifies it

data (BTM, sedol), and produces an output file showing for every event firm up to 5 matching firms in the same industry group. The matching between firms is based on BTM information.

25 size / book-to-market portfolios: This program takes as input a number of years (usually 4 years), starting and ending years, event firm data (sedol, event date). Then for each year between the starting and ending years, it prompts the user to enter BTM data (sedol, BTM) and the file of excluded firms for the given year. The program produces 34 output files. 25 output files are produced for 25 portfolios formed by arranging firms into quintiles according to their BTM (book to market ratio) and market capitalization value. 4 files for BTM firms for year i sorted by BTM. 4 files for BTM firms for year is sorted by MCAP. One file for all event firms with their 3 years returns (starting from event date backward or forward), the portfolio in which the event firm is allocated in the event year, and the return of all portfolios of the same quintile of the event year for 3 years.

Fama-French three factors model:

(1) Portfolio formation: This program takes as input excluded firms data (sedol), returns data (LSPD RETS), and BTM data (BTM, sedol), and produces 10 output files. Two output files are for sorted firms (firms in BTM file except excluded one) by BTM and market capitalization. Six output files (portfolios) are grouping firms allocated as big/small MCAP, high/low/medium BTM One output file for all firms in BTM files (ignoring excluded firms) with all necessary information about the firm. One output file for portfolios returns calculations.

(2) Market returns: this program takes as input a given year and excluded firms in this year (sedol num), and produces an output file of all firms in LSPD (excluding firms from the excluded input file) with their monthly returns for the input year. The output file also includes calculation of equally weighted average returns of all firms and average returns of all firms weighted by market capitalization.

Bootstrapping: this program takes as input a vector (array) of float numbers and produces n re-samples of b sub-samples of the vector of input data. In the program, n is taken

as 1000 and b as 500. Some t-statistics are computed for the re-samples.

Informal statistics on the scale of programs written are summarized in the table below. They refer to the four programs implemented: the BTM and MCAP matching, the 25 portfolio, and the Fama and French portfolios (FF3) programs.

	MCAP MATCHING	***BTM MATCHING***	***25 size / book-to-market portfolios***	***FF3***
Number/ Size of input files	3 input files events firms (2-3kb), excluded firms (2-3kb), RETS (45Mb)	4 input files: event firms (2-3kb), excluded firms (2-3kb), RETS (45Mb), BTM file (30kb)	9 input files: event firms (2-3kb), BTM file and excluded firms files are input for 4	3 input files: BTM file (30kb), RETS (45Mb), excluded firms (2-3kb)
Number/ Size of output files	1 output file (300kb)	1 output file (250kb)	34 output files: * 25 portfolios * 8 files (BTM, MCAP sorting) * event firms and their similar portfolios returns in 4 yrs	10 files * 6 portfolios * sort by BTM * sort by MCAP * return calc
Max no. of times a single file is opened	12 times (RETS)	7 times (RETS)	3 times (RETS)	8 times (RETS)
Running time	~20min	~20 min	~ 1 hour	~15 min
Size of program ***Lines/bytes***	42kb 1585 lines	25Kb 993 lines	34 kb 1415 lines	24kb 1073 lines
Basic blocks	Find min & Sequential search algorithms	Find min & Sequential search algorithms	Bubble sort and sequential search	Sequential search and bubble sort
Rules / Conditions/ data structure	2 user defined types are declared: (1) for a Firm from RETS file - LSE firm, (2) for an event firm and its associated information.	3 user defined data types are declared: (1) RETS firm, (2) event firm, (3) firm from BTM file	four-dimensional array (year, portfolio i, j, nth return)	3 user defined data types are declared: (1) RETS firm, (2) event firm, (3) firm from BTM file
Frequency of usage per study	Used per event year/event study (9 times for 9 yrs) 9*3 =27 for 3 types of event studies	Used per event year/event study (9 times for 9 yrs) 9*3 =27 for 3 types of event studies	Used per event year/event study (9 times for 9yrs 9*3 =27 for 3 types of event studies	Used per event year/event study 9 times for 9yrs 9*3 =27 for 3 types of event studies

Table 8.2 Preliminary statistics on the scale of programs written

These statistics reveal intensive file processing and data manipulation operations. Memory capacity shaped the structure of the programs. The amount of data that can be held in memory as n dimensional arrays of user defined data type is limited. This constrains the way user-defined data types are declared and increases the need for file processing operations. For example, the LSPD returns file was opened 12 times in the matching program to search for up to 5 matching firms (MCAP / Industry matching or BTM / industry matching) for a particular event firm.

The above statistics inform the development of computer-based tools for data intensive financial analysis. Such tools should take into consideration the amount of data processed, speed of processing, data structure used, and data manipulation paradigm.

❸ Results analysis:

The research conducted by Ho and Abhyankar has so far examined the abnormal performance of a large sample of UK firms following rights issues and placings over the period 1989-1998. The research findings revealed in [AH01] show that long-horizon abnormal performance in the UK financial market is highly sensitive to the methodology used to compute the abnormal performance depending on the choice of benchmark. This confirms the view that the apparent anomaly of under-performance[12] may be due to the methodology adopted, and the long-horizon abnormal return may disappear with a reasonable change in techniques. The study concludes with the need to adjust correctly for risk in measuring long-horizon security returns.

The financial research development cycle (FRDC) has been framed within a web-based environment serving a documentary purpose. The PPP (Pre-Publish-Push) technique has been used to develop this web environment. The web environment contains collected UK data, a description of the techniques used, the programs developed, and access to related publications (cf. [MaadWeb02]).

[12] There is mounting evidence, from USA and Japanese markets, of long-horizon security price under-performance following corporate events like initial public offering (IPO) and seasoned equity offering (SEO).

8.2 EM Technology Support for the FRDC

Chapter 4 considered the broad implications of EM for computational activity in an application context, and for software system development in particular. This is manifested in a radical shift that supports openness, situatedness, and context dependent software system development where modelling (as opposed to programming) is central and user-developer-designer situated collaboration is essential. This shift potentially enables a closer integration of the software system development activity and various activities undertaken in real world domains.

EM technology motivates a radical shift in software system development. This gives EM technology a greater potential to establish a closer integration between the Finance Research Development Cycle FRDC and the software system development activity. Such integration can potentially enrich the FRDC with greater computer-based support and a better understanding of the role of the computational activity in financial research. The following sections discuss the relevance of the EM paradigm shifts in SSD to the computer-based support of the FRDC. Three issues are considered:

i. collaboration (user-developer-designer / social network of finance researcher)
ii. the amethodological approach to software system development in EM
iii. the shift from programming to pervasive agent-oriented explanatory modelling

i. Collaboration

Collaboration in a distributed modelling environment is an important feature in providing computer-based support for the FRDC. In the context of the Ho case study, this is illustrated by various observations about the user-developer-designer interpersonal interaction that show that software system development is a social activity that needs computer-based support to meet its objectives.

In the Ho case study, the software system development process involves the designer (Keng-Yu Ho), and the developer (the author). The designer is also the user.

The major challenge faced in the software development process is the gap between the designer's knowledge of the meaning of the financial data set, and the abstract representation adopted by the developer to represent this data set in the program. As an example to illustrate this gap, consider the 12 monthly returns in a given year for a firm. In the designer's mind, specific year and specific month under consideration are important. For the developer returns

are represented as an array of float, and every return is accessed by its index. It is not important to the developer whether this index refers to the month of January or February or any other month within a particular year so long as the returns are abstracted as a fixed size array and accessed through an index. The gap between the designer's and the developer's views is widened as new requirements arise that attach greater importance to the real world meaning of abstract variables in the program. Close interaction between the developer and the designer reduces this gap to a certain extent, and sometimes this interaction takes the form of direct intervention from the designer in the software development to dictate a relationship between variables.

Turning a programming activity into a shared modelling activity and adopting a modelling paradigm that establishes a stronger virtual correspondence between real world entities and their virtual representation helps in reducing this gap and its serious impact on the financial research development cycle.

The following figure illustrates the conceptual gap between the designer (financial expert) and developer (programmer) in providing computer-based support for the financial research development cycle. This gap can be bridged by efficient communication aimed at establishing a greater correspondence between the state of the financial research and its corresponding computer-based model. This state has four components (cf. [BRWW01]): visible external, hidden / internal, mental, and situational. An example of a visible state is the value of financial indicators presented in tabulated output formats. An example of an internal state is the abstract computation that determines the visible state of the financial indicator. An example of a mental state is the meaning and financial relevance of the financial indicator when used in the analysis. An example of a situational state is the context of a particular adopted methodology.

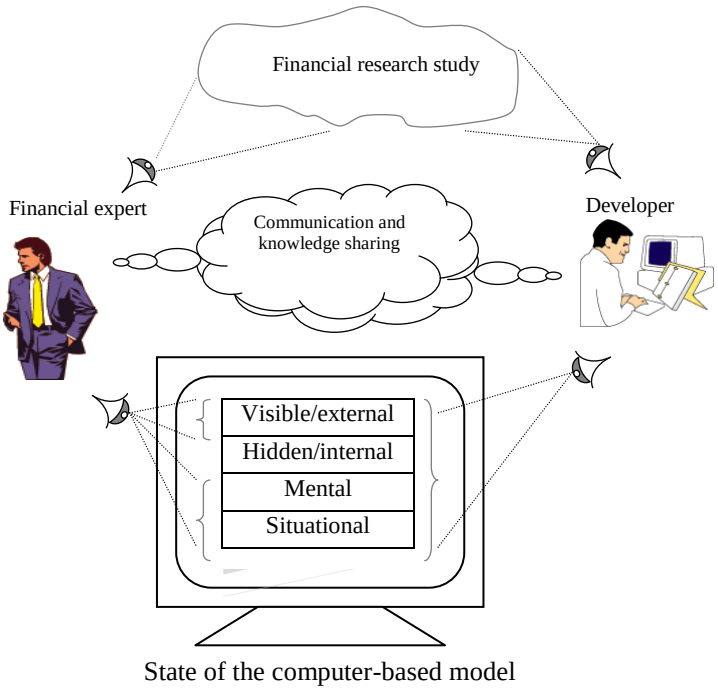


Figure 8.2 Sharing knowledge about the state of the financial research and its computer-based support

ii. amethodological software system development

An amethodological approach to software system development promises a closer integration between the software system development activity and the finance research development activity. In the context of the Ho case study, this emerges from a closer consideration of the FRDC and its associated software system development. A situated account of the FRDC would be better supported by a non rigid software system development approach.

A typical life cycle of a software system may be divided into seven stages: problem definition, feasibility study, analysis, system design, detailed design, implementation, and maintenance. In relating the considered finance research development cycle (FRDC) to a software system development cycle (SSDC), there is no direct one to one mapping between the different stages in the two cycles (cf. Figure 8.3). The financial data collected and the structure, meaning, and uses of this data play a major role in the problem definition, analysis, and design. The appropriateness of a particular econometric approach can only be tested after doing a complete analysis of the results. This is a major concern, as much time and resources

can be wasted on implementing econometric approaches that do not support the expected conclusions effectively. The problem definition is dictated by the designer. The feasibility study considers different algorithms for implementing the suggested econometric approaches. The developer's role is important in the implementation stage, but the designer has more responsibility and influence on the direction and evaluation of the work. The software system is used throughout the whole life cycle of the FRDC and terminates at its end. Unlike a SSDC, where the user enters at the end of the testing stage to use the finished software product, the user in the FRDC is the designer and uses the evolving software product throughout the whole development cycle. Once the final stage in the FRDC is reached, the software product is no longer useful[13]; it serves the user incrementally during the development cycle. The nature of specialized finance research is such that it is difficult to make use of the special purpose software product developed throughout the whole FRDC in subsequent specialized research without much amendment. This raises the issue of re-using general purpose utilities to support future similar research.

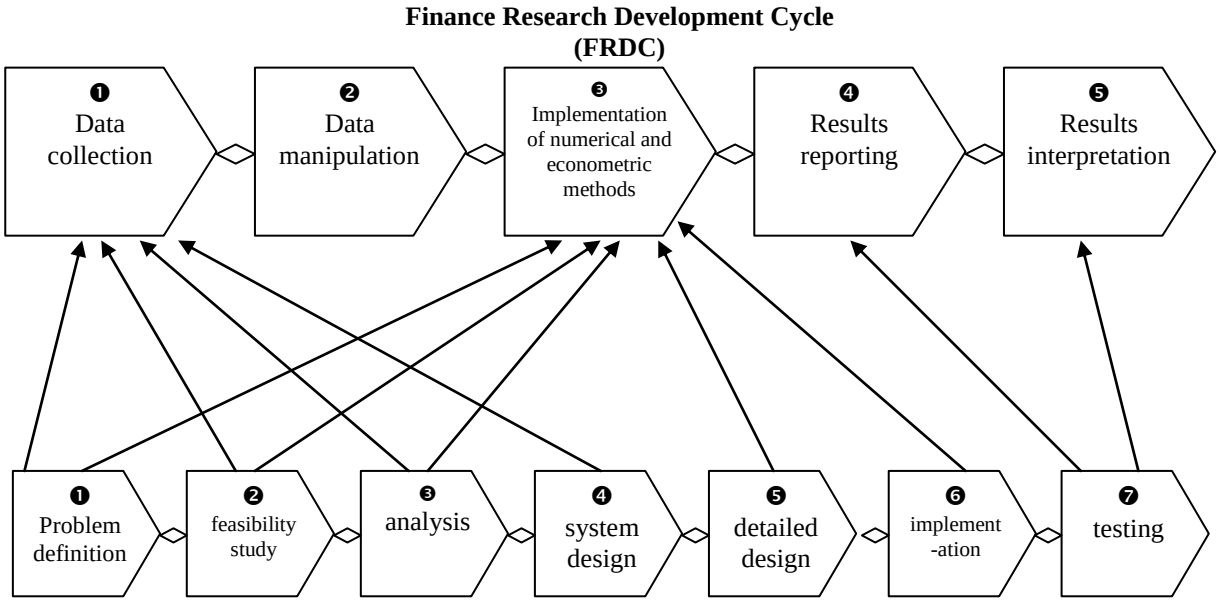


Figure 8.3 FRDC Vs. SSDC

[13] Except as reference to future similar research work, that might adopt different methodologies with similar spirit.

One of the major problems encountered in the FRDC is the need to find reliable techniques for testing. Failure to get the expected results might be attributed to the inadequacy of the adopted methodology, an incorrect implementation of the methodology, or inappropriate processing of the resulting output after the implementation stage.

The above account of the FRDC reflects the author's view, but the designer might well have a different FRDC model in his mind that is not communicated to the developer. This serves to emphasise how the specialized nature of the research makes it difficult to replace the role of the human designer and developer by an intelligent agent. It also serves to highlight how ambitious are the aspirations of logicist AI technology. It is hard to conceive the encoding of specialized financial research knowledge that could inform an intelligent agent capable of collecting, organizing, exploring different methodologies, selecting an optimum methodology, implementing it, analysing the results and inferring conclusions.

iii. The shift from programming to pervasive agent-oriented explanatory modelling

A situated account of the FRDC and a closer integration between the finance research development and software system development activities motivate a paradigm shift from programming to pervasive modelling in providing computer-based support for the FRDC. Whereas programming is limited to the computer-based implementation of methodologies and algorithms using third and fourth generation languages, pervasive modelling contributes to a more holistic view of the FRDC that integrates the human and the automated activity.

Current approaches to computer-based support for the FRDC (provided by third and fourth generation languages, Object oriented, Virtual Reality, and Artificial Intelligence technologies) is limited to the programming / implementation of algorithms and methodologies. The following paragraphs discuss several of these approaches with reference to the Ho case study. The aim of this discussion is to emphasise the importance of providing computer-based support for modelling the FRDC that takes into account the wider scope involving the support of:

- end-user programming
- greater human computer integration
- semi-automated activities
- experiential knowledge construction
- the generalisation of the spreadsheet concept beyond data in tabulated cells
- the user - developer - designer collaboration

The OO approach to support the FRDC

The concepts of modularity[14] and abstraction[15] [Win93] that motivate the object oriented programming approach were little considered in the Ho case study for two reasons. In the first place, many of the operations on the user defined abstract data types could not be preconceived and it would have been difficult to identify all the methods for a particular object at the outset. Second, the programs written are tailored to their highly specialized research purposes to such an extent that addressing modularity and re-use would have been an unnecessary distraction that delayed the delivery of essential results. A good example of the application of object oriented programming in the area of finance is in automating bank operations [Win93]. The operations on a bank account are limited and well defined. This makes it appropriate to declare an abstract object data type **`bank account`**, and to specify the methods of this object as a constructor, a withdraw, a deposit, and a balance method. In the Ho case study, different econometric and numerical algorithms are implemented as the need arises and as particular algorithms fail to provide the expected results. This fact makes it difficult to have an *a priori* description of objects.

The author did not optimize the algorithms used in the implementation with respect to speed and efficient use of memory as the designer's major concern was to get the correct output for a given input and process. It might be that object orientation brings more modularity and manageability to the programming code, but this issue was outside the scope of the author's investigations.

Exploration vs structured methodologies in financial analysis

Adopting structured numerical and econometric techniques to analyse large financial data sets is a popular approach in academic financial research. Uncovering hidden knowledge in huge data sets is a challenging task, and structured analysis has a primary role in gaining knowledge from large data sets. For a single case study, different numerical and econometric techniques can be applied. Each of these techniques reveals a hidden knowledge aspect in the data set. A major problem that might be encountered in financial research is the failure of a particular econometric or numerical approach to deliver the expected result. Some measures could be taken to prevent the commitment to an inadequate approach or to an incorrect implementation. A major concern is that the time spent to explore the adequacy of a particular

[14] Modularity ensures that all elements of the system have well defined boundaries separating each other and clearly defined interfaces through which the elements interact - the modules only interacts through interfaces.

[15] Abstraction is the act of ignoring the detail of a component of a program and reasoning only about the interface. Two types of abstraction exist: data abstraction and functional abstraction.

econometric or numerical approach might be the same as the time for implementing and testing it. This motivates an exploratory approach in analysing large financial data sets.

End-user support throughout the FRDC

In considering the use of end-user programming[16] in the implementation stage of the RDC two limitations are confronted. The first limitation is the large amount of file processing needed to implement the selected methodologies. The second limitation is the need to perform multiple repetitive operations that are not available as built in functions and procedures in end-user programming tools such as Gauss and Matlab. A solution emerges with the evolution and progress of the RDC. This solution would be to develop generic utilities for re-use by other similar research. In the context of the Ho case study these generic utilities might include matching, portfolio creation, market return calculations, and bootstrapping. The possibility of implementing these generic utilities as encapsulated objects is a good solution for re-use and modularity. The use of visual queries techniques to manipulate large data sets might be a complicated approach in specialized financial research such as the one considered, due to the complexity and diversity of special purposes data manipulation techniques.

AI support for the FRDC

In considering the application of an AI approach to conduct the Ho case study, two questions are addressed:

i. Can AI techniques (neural networks, fuzzy logic, visual exploration using self organized maps) replace / complement the econometric / numerical approaches adopted?
ii. Is it possible to encode the knowledge constructed throughout the whole FRDC, and use this knowledge to create an intelligent agent capable of replicating the same analysis in a different context (different country)?

Though the answers to these questions that follow may appear superficial, they raise deep conceptual issues relating to the success of AI technology in actively supporting advanced and specialized financial research.

i. The literature overview conducted by Ho (2000) reveals that different econometric approaches lead to different inferences and conclusions. This calls the viability and reliability of econometric approaches into question and leads the author to conclude that replacing an econometric technique by an AI technique is not a solution. A complementary role is rather advocated to get satisfactory results.

[16] The terms 'end-user programming' and 'programming using fourth generation languages' refer to a programming style that gives the end-user greater computational assistance whilst requiring minimal knowledge of the complex aspects of programming.

ii. It is hard to imagine encoding the designer and developer expertise used throughout the whole RDC in a knowledge base that can be used in a logicist AI approach to automatically re-define a similar problem, explore a solution set and select one optimum solution. The first challenge is the difficulty in encoding the expertise expressed in a group activity (designer, developer collaboration) in defining the problem, selecting an optimum methodology, implementing this methodology and testing the results. Collaborative social activity is rich in interaction and knowledge sharing. AI techniques would be better targeted at supporting the collaborative activity rather than attempting to encode and automate it. The second challenge concerns the context and situation of the financial research. There can be no guarantee that distributed intelligent agents capable of taking autonomous actions to conduct a UK based financial study would be effective in carrying out a similar study in a different market.

VR support for the FRDC

Traditional application of VR technology in finance attempts at merging the mechanism of a trading process with the analysis of the results of the trading process in a synthesised 3D visual environment (cf. http://www.nyse.com/floor/floor.html). This highlights the importance of the human visual system and its role in supporting knowledge construction about an environment.

Where case studies such as the Ho case study are concerned, 3D visualization in a virtual scene may be helpful in navigating large data sets in an exploratory way. Eyeballing techniques potentially have a useful role in conducting or supporting advanced financial research analysis. In the Ho case study, the nature of the data set analysed (event date, monthly returns over a particular window, particular indicators – market capitalization and book to market ratio) and the required operation on the data set (matching, sorting, portfolio allocation, bootstrapping) indicate the need and importance of relying on complex methodologies to conduct the research. This makes eyeballing techniques, that are enabled with VR technology, useful at an end stage. Their role would be in enhancing the visualization of results and in supporting their interpretation.

Another role for VR technology would be to support the description of the event occurrence mechanism. Such a description is found in [AH01], where the sequence of activities followed by UK firms seeking capital through rights issues (a corporate event) is described in the course of reviewing particular aspects of the institutional framework in the UK market.

The above discussion of various approaches to supporting the FRDC reveals that the computer-based support offered by current technologies is confined to implementation and

reporting stages (cf. Figure 8.1). In this kind of support, the computer plays the role of an advanced calculator. The cognitive activity of the financial expert involved in stages 1, 2, and 5 (data collection, manipulation, and results interpretation) of the FRDC enjoys no automated support apart from routine data processing. This is despite the fact that this cognitive activity, if appropriately conducted and supported, would have great impact on the success and usefulness of the computer-based support for stage 3 and 4. This emphasises the importance of modelling the FRDC in a more holistic manner.

With the adoption of an EM approach to SSD in this context, the modeller can bring understanding of human agency and potential computer agency to bear on the construction of a computer-based artefact. Human agency relates to the cognitive activity associated with the choice of strategies for data collection and manipulation, the methodologies for financial analysis to be implemented, the style of result reporting that impact on the interpretation of results. EM can potentially enable the modeller to embody these activities in an artefact without invoking circumscription. This allows the SSD to proceed without premature commitment to a particular data management strategy, analysis methodology and reporting style. This feature, together with the agent-oriented qualities of the artefact and its explanatory character, potentially offer flexibility for experiment that can transform the scope for the modeller both to frame, and to adapt to, new requirements.

By way of a motivating illustration, in the context of the Ho case study, the organization of data files (**`event firm`**, **`book-to-market ratio`**, and **`excluded firms`** data files) and the choice of benchmark (matched size / industry, matched book-to-market ratio / industry, and portfolio benchmarks) used in assessing the abnormal performance of a firm, involve strategic decisions to be taken by the financial expert that have radical implication for the software requirements of the FRDC and are subject to revision.

8.3 The Implications of data intensive financial applications for EM Technology

Adapting EM technology for data intensive financial applications raises issues related to ***referencing***, ***processing***, and ***interpreting*** large volumes of data. Research in progress undertaken by EM technology to tackle these issues is overviewed below.

8.3.1 Referencing large volumes of data

Three approaches have been used to address the limitations of definitive programming within the Empirical Modelling Framework (EMF) in referencing large volumes of data and porting EM concepts to an object and system level:

The first is by applying the concept of dependency maintenance to explicit relationships between abstract objects data types. This is facilitated with the use of JaM[17] the Java Maintainer application programmer's interface (API). As documented in [Cart99], JaM provides a generic dependency maintenance system for use in object-oriented programming. It also provides facilities for organising the information in a dependency structure into virtual directories and manages user and group permissions to access and modify the data and dependencies. JaM's dependency maintenance can be considered as a generic spreadsheet in which the *cells* of the spreadsheet are instances of any class chosen by a developer that implement an interface of the API, not just conventional types of dates, strings and numbers. The dependency between JaM *cells* is established by the method of a class implemented by a developer that is equivalent to a spreadsheet *function*, such as "SUM". Change the value of any of the values in the *cells* and the values of all dependent *cells* are automatically updated. In contrast to a spreadsheet, items of data do not have to be organised in a grid and are identified instead by a name and directory path. Many *cells* can be changed simultaneously and JaM is fully multi-user and threaded, taking advantage of multiple CPUs if they are available.

The second approach used in addressing the limitation of definitive programming in referencing large volumes of data is by using virtual agency [Sun99]. Conceptually, virtual

[17] Under development by Richard Cartwright at BBC Research and Development, UK. The Java Maintainer Machine API version 1 (JaM) was developed as a demonstration of a generic object-oriented dependency maintenance system illustrating a new parallel method for definition update, the DM model. Developed in 1996 using Java 1.0, JaM is documented as part of the Phd thesis [Car99].

agency provides a way of attaching a definitive script to a particular observer, typically so as to represent the personal perceptions of that observer. Ideally, the association between a script and its observer should be defined and manipulated as a form of dependency, however, practically virtual agency is managed in dtkeden in a procedural fashion. With the virtual agent notion, a definitive script can be used as a pattern to generate different definitive scripts associated with different contexts. This allows reusability of a definitive script in dtkeden. Another solution for re-use introduced in [Sun99] is the introduction of Generic Observables (GO). Unlike those observables that correspond to real world objects in the modeller's external environment, GO are created to correspond to the modeller's experience, which is inside the modeller's mind and emerges from repeated description of certain observables with the same characteristics.

The third approach to addressing the limitations of definitive programming in referencing large volumes of data is distributed modelling. Distributed modelling [Sun99, Car99] helps in spreading the data processing load on different workstations. This increases the capacity of the distributed model to handle larger data sets describing a generic observable.

8.3.2 Processing large volumes of data

The main tool used in EM research so far is the **EDEN** interpreter and its distributed variant **dtkeden**. The **EDEN** interpreter is primarily a dependency maintainer for formula definitions and action specifications [Tru96]. The main weakness in all interpreters, including **EDEN**, is that program instructions and data when parsed line by line, translated and executed, must be held in main memory all the while. In general, interpreters are effective and efficient for applications that requires intensive processing on a small set of data. They are not so effective or efficient for applications that perform a small amount of processing on large volumes of data. Such type of applications are commonly referred to as *data intensive* applications (as opposed to *processing intensive* applications). **EDEN** inherently keeps all of the large volume of data within memory, while performing little or no processing on them. This tends to use up system resources and reduces the execution speed of programs. This creates a serious problem when considering data intensive applications. Attempts at providing a solution to the problem of increasing the efficiency of data intensive **EDEN** applications and at providing a means to manipulate large volumes of data effectively while respecting the EM philosophy of definitions and evaluations based on dependencies, was initiated by Truong

(1996). His solution was to design and implement a Relational Language Interpreter **EDDI** and to use it with ORACLE database. This interpreter forms a front end to **EDEN**, where relational instructions are translated to database manipulations. **EDEN** script can include action specifications that can simulate the storage, retrieval and manipulation of data. The objective is to develop data intensive applications in **EDEN** while maintaining the values of an application variable in a database. When the necessary values are actually needed, then they can be retrieved from the database into memory variables for manipulation. Hence, large volumes of data do not need to be kept in memory, freeing up memory resources. The relational language interpreter is designed with the relational data base model in mind. The model deals with tables of data and operations on data tables.

The extent to which JaM and dtkeden offer enhanced processing capacity has already been remarked. A more radical development, aimed at improving processing efficiency by handling dependency maintenance closer to the hardware level can be found in the definitive assembler machine, as introduced by Cartwright [Car99]. There will always be processing overheads attached to maintaining dependencies that can be eliminated once the functionality of models has been circumscribed for specific applications. This has been the theme of projects directed at extracting conventional programs from ISMs, as discussed in [ABC98]. Translation activity of this nature will always be desirable – if not essential – for the effective application of EM in data intensive and performance critical applications.

8.3.3 Interpreting large volumes of data

Since many data sets fail to satisfy the underlying assumptions of most econometric approaches, unconstrained visual exploration is gaining more importance in the financial industry. EM technology supports the interactive visual exploration of data sets [Roe00][18]. The aim of the research undertaken by Roe (2000) is to overcome the limitations of conventional enhanced database approaches to exploring large data sets. These include: the need to preconceive the field(s) to view before writing a database query; the problem of uncovering hidden relationships between fields when a large volume of data is returned by a query; the difficulty in spotting complex relationships that might exist across multiple fields in numeric data. Roe's proposed visualization is extrinsic (the properties of the data elements to be displayed are mapped to a physical counterpart – shape, color or texture). The visual

[18] The work was inspired by the Attribute Explorer originally developed by IBM, UK, due to an idea of Bob Spence, Imperial College, London.

exploration environment developed is regarded as an interface between the human and a data set. The relationships that may exist in the real world between the data attributes are virtually mediated to the computer environment as visualized dependencies. Human insight guides the exploration of the data set in many personal subjective ways. This helps, at an initial stage, in apprehending the data and its meaning. Compared to statistical regression, visual exploration helps to provide a deeper understanding of complex non-linear relationship between data. The challenge of interactive visual exploration is the difficulty of formulating in mathematical relationships our experiential knowledge of the data as apprehended. This motivates future research on transforming visual and mental understanding into formulated symbolic knowledge.

8.4 Summary and Outlook

This chapter considered a case study related to data intensive financial analysis research. The analysis of the case study was enriched by EM principles that suggested a situated account of the financial research development cycle. At the same time, EM technology can benefit knowledge about the requirements of computer-based support for data intensive financial analysis from the case study.

The future research agenda for EM technology to address the needs of computer-based support for data intensive financial analysis comprises technical and conceptual objectives including:

(a) Further research to integrate EM technology with database technology and VR technology. This is considered to be a long term objective.

(b) The application of the principles of observation, agency and dependency for exploring, in an amethodological way, hidden patterns of relationships in large financial data sets. This can be referred to as "amethodological exploration" of large data sets or "amethodological data mining". Data mining of this nature is led by the modeller's insight in exploring a virtual world of abstract financial indicators. Visualization of state is an important issue. Challenges that may be faced by the proposed amethodical exploration of large financial data sets are: (1) the difficulty of transforming experiential knowledge, gained in the course of exploration of hidden patterns in large data sets, into objective mathematically formulated knowledge; (2) the difficulty in reconciling an amethodological exploration of large financial data set with a

well established econometric approach; (3) the difficulty in resorting to statistical analysis on the results of the amethodological exploration.

Chapter 9

Conclusion

9.0 Overview

This chapter summarizes the research presented in the thesis, highlights its limitations, and proposes further work to support the penetration of EM technology in the finance domain.

9.1 Research Summary

The research presented in this thesis was initially motivated by the author's interest in exploring computer-based support for finance. The author was attracted by the potential of EM technology to deliver a framework supporting primitive observation and experimentation in the financial domain. This is essential in construing phenomena that are difficult to capture in abstract mathematical models or to understand by applying econometric and logicist AI techniques on historical data sets.

The survey conducted on the finance domain and the application of computer-based technology in this domain reveals special characteristics and a diversity of tools and technologies. The finance, the computer and the communication industry are witnessing continuous change. It is difficult to identify exactly the drivers of change and it is clear that the pace of change will continue in the future.

"The only thing that won't change is the fact that everything will continue to change".

Timothy W. Ryan [Rya97].

The future will see further integration of computer tools, convergence of technologies (computer, communication, TV), new financial instruments, increasing financial market integration, new structures and new roles for participants in the financial market.

To suppose that all change can be handled by delivering a recipe is to fail to do justice to the true meaning of the concept of change. Experiencing change and describing this experience, in more human terms, though it may not lead to an efficient or an exact solution, can assist in coping with change without imposing a structure upon it.

Managing and coping with change is not straightforward. We need to understand this change, its implication, and its future direction. Describing change in a formal framework simplifies the reality behind change and hides its complexity. We need a framework that addresses the complexity of change without circumscribing or making erroneous assumptions about the pattern of this change. Be it broad human thinking or a broad computational framework or a combination of both, embracing the pace of change remains a big challenge for the human and the machine.

The thesis aims at addressing the need of a wider agenda for computing in the finance domain. It explores a computational framework, different in character from formal highly structured ones, and that can potentially cope with change with high flexibility and adaptability. This computational framework emphasizes situated actions, the human role

(including its weaknesses), and human machine collaboration whilst looking deeply at the requirements and implications of change.

The thesis has identified key issues for the application of computer-based technology in finance. These issues motivate a wider agenda for computing in finance. A paradigm shift at the computational level is needed to meet this agenda. EM technology has application and future prospects in the finance domain and can potentially meet this agenda's requirements.

The thesis has considered four case studies for the application of EM technology in Finance. The main findings of the research may be summarised as follows:

- ***Identifying key issues for computer-based technology in finance***

  Key issues for the application of computer-based technology in finance are centred around software integration and virtual collaboration in the financial enterprise, the shift from the old to the new trading model in the financial market, computer mediated interpersonal interaction in financial engineering, and software system development for the financial research development cycle.

- ***Framing the wider agenda for computing in finance***

  The wider agenda for computing in finance addresses technical and strategic demands that can be met by adopting a broad foundation of computing drawing on a suitable framework for deploying prevailing computing practices and leveraging novel uses of the computer within this framework. Technical demands take into account the experiential and the human factors, and the pervasive[1] emergence of computing. Strategic demands take into consideration qualitative uses of computer-based technology to support diverse real world activities that are cognitive rather than operational, such as decision support, business process modelling, learning, management, interpersonal communication, etc..

- ***Motivating the paradigm shift at the computational level***

  Addressing the wider agenda of computing motivates a paradigm shift at the computational level that involves a reconstruction of the software system development activity in a wider framework capable of addressing technical and strategic demands. This involves the emphasis on amethodological, situated and human centred approaches to software system

[1] This factor refers to the application of computer-based technology in various devices including interactive TV, mobile phone, etc.

development that favour user-developer-designer collaboration in distributed modelling of a broader view of a system where human role is central.

- ***EM applications and prospects in finance***

The main contribution of the thesis has been to propose several applications of EM in finance and to provisionally evaluate their prospects. In particular I have shown how EM technology can potentially address the wider agenda of computing in finance by:

i. *considering software integration in the financial enterprise as a social and technical activity. A situated integration model emerges drawing key principles for software integration taking into account the interaction of various human and electronic agents*[2].
ii. *considering virtual collaboration in the financial enterprise as a situated context dependent activity within a social network where every participant has its digital and non-digital information horizon. This motivates greater flexibility, adaptability, and human integration in the development of virtual environment for collaboration.*
iii. *providing a computer-based support for the shift from the old to the new trading model. This consists of developing a computer-based artefact The Open Financial Market Model (OFMM). The construction of this artefact takes into consideration various factors including:*
    - *experiential rather than formal approaches to knowledge construction*
    - *the role of agency and observation for a richer description of interaction in the financial market*
    - *the semantic relationship between the computer-based artefact and the real world financial market context*
    - *openness*
    - *flexibility*
    - *modellers' subjective personal insight and perception*
    - *a closer integration between the computational activity and various financial market related activities including: learning, decision support, modelling the trading process, etc..*
    - *semi automated activities*
    - *collaboration amongst modellers involving the sharing of private / subjective and public / objective perspectives*
    - *evolving requirements engineering*

iv. *supporting computer mediated interpersonal interaction in financial engineering through the communication of definitive scripts and the creation of the context of interaction based on agency and dependency. This takes into consideration:*
   - *the role of agency (privileges to take actions) in the communication*
   - *the diverse modes of interaction*
   - *the collaboration of participants in a classroom or practitioner use context.*

v. *considering the need of a closer integration of the software system development activity and the financial research development cycle. This draws on:*
   - *amethodological non-structured approach to software system development and a situated account of the financial research development cycle (FRDC)*
   - *user (financial expert) – developer (programmer) – designer (financial expert) collaboration*
   - *a holistic view of the financial research development cycle based on modelling rather than programming*

The above preliminary findings foresee a promising role for EM technology in the finance domain. This encourages the adoption of the principles of observation and experimentation to construe financial market phenomena to complement current existing approaches. With the openness and situatedness of the software system development activity adopted within the EM framework, the designer (financial expert)-developer (computer expert)- user (with general expertise) as well as different financial market participants guide the evolution of the financial system in response to change and to their experiential knowledge and insight in understanding the change. In this manner, the financial and computational activity integrate more coherently, while preserving the role of different participants and respecting their domain knowledge.

[2] The term agent refers to entities capable of inducing state change to the integrated system.

9.2 Research Limitations

In considering a wider agenda for computing in finance (enterprise, market, and investment levels) some research limitations need to be acknowledged. These are attributed to three factors: time and resources, technology support, and research infrastructure.

- Time and resources: every research has time and research limitations. The author acknowledges that more practical work should be undertaken to further support the application of EM technology in the case studies considered viz: software integration; virtual environments for collaboration; the open financial market model (OFMM); data intensive financial analysis in general and the Ho case study in particular; modelling financial instruments in general and the affine term structure model in particular.
- Technology support: every technology has its own limitations, including the proposed EM technology. In general, technology limitations can be attributed to three gaps: the gap between the principles / foundations and the techniques of the underlying technology; the gap between the techniques and the features of the tools implementing the techniques; the gap between the principles / foundations and the corresponding tools. These gaps are apparent in virtual reality (VR) technology, artificial intelligence (AI) technology, Object Oriented (OO) technology, and Empirical Modelling (EM) technology.
- Research infrastructure: the author faced a lack of supporting infrastructure to conduct the research at an academic and practitioner level.

The following paragraphs elaborate on each of these limitations, detailing where possible the essential reasons for these limitations.

• Time and resources limitation

This research has aimed at investigating the potential application of the Empirical Modelling technology to the finance domain. Time and resources constraints imposed some limitations on the thesis:

i. Focusing upon the thesis aim, some aspects of the EM technology and the finance domain were overlooked:

EM technology has great prospect of application in the area of geometric modelling, engineering design, and computer aided manufacturing. These areas of applications of EM

were not sufficiently addressed in the thesis, especially the area of geometric modelling which could have enriched the development of the OFMM.

ii. The title of the thesis suggests applications and prospects of EM technology in finance. The considered applications and the foreseen future prospects of EM technology in finance could be further supported. Where principles and foundations are proposed such as in the case study of software integration, virtual environments for collaboration and the OFMM, more practical work can be undertaken to support these principles. Where a potential prospect of EM technology is emerging, further theoretical and technical work is needed.

iii. This research has not involved a detailed comparison of the EM technology with other technologies used in the finance domain. Although a background review of the application of different technologies in finance is provided in chapter 2, the application of these technologies to particular areas in finance is not compared explicitly with the application of EM technology to these areas.

iv. Industry contact is essential for the validation of the application of a particular technology to a practical domain, like finance. The author made several academic contacts in the field of financial studies, but did not pursue many industry contacts.

• The technology limitation

In [BWM01] several limitations of EM tools are acknowledged. EM tools serve as a proof-of-concept for the application of EM principles in many different domains. EM tools need better visual interfaces, additional script management features, enhanced end-user computing, enhanced data storage capability, and further integration with other technologies such as VR and web technologies.

Further research is needed to assess the qualitative and quantitative contribution of EM technology to different fields of study.

EM literature does not direct much attention to key management and planning concepts such as optimum resource allocation (budget, time, personnel). Speed, efficiency, low cost, professional design are not major themes in EM technology. This fact makes EM technology more suitable at an academic and research level. For industry penetration and approval, Empirical Modelling technology should integrate resource management in its application to different fields of study. Conventional technologies are predicated on optimization, so that addressing optimization is strongly connected with relating EM to conventional computational approaches [ABC95].

Within the Empirical Modelling Framework, the boundary of the development process is not limited. This is an advantage and a disadvantage at the same time. It is an advantage because the development of software or products will always benefit from evolving knowledge. The disadvantage is that the product being developed might never reach a final stage and cannot be optimized for unspecified functions.

• Limitation of the research infrastructure

The research has been inhibited by a lack of a supporting research infrastructure, both at an academic and practitioner level. As mentioned in earlier chapters, the computer implementation of different techniques and methodologies used in financial research is rarely documented in the research itself or separately. This can be mainly attributed to the importance of result reporting and analysis in finance, which takes precedence over the computer implementation. Financial experts are not much concerned to reveal their adopted technology support as this will not give any value added to their results and inferences. AI technology is currently a pioneer technology in bridging the gap between financial analysis and computer support to this analysis. The trend is towards more mechanisation of the financial reasoning process, and towards trying to treat information related to financial markets in a similar way to the results from experiments in a physics laboratory. Methodology, performance, and speed come first in the agenda for the use of the computer in finance. Experiential knowledge, situated activities, semi-automated activities are not yet explored in the domain of the application of IT in finance. This fact makes the research harder, as it is attempting to give greater weight to links between the domain of computer and finance that favour the social and situated aspect of the financial discipline. Though these links exist, they have not so far been given much consideration.

9.3 Further Work

In exploring the application of EM technology to the finance domain, five case studies are considered. The theoretical and practical application of the EM technology to each of the considered case studies can be reinforced along the following guidelines:

(1) The Situated Integration Model (SIM):

In addressing the problem of the integration of ERP and e-commerce application, the Empirical Modelling approach can meet short-term and long-term integration requirements:

In the short-term an SIM as introduced can potentially complement orthodox approaches to integration in many ways. Our proposals draw upon previous research into the role of EM in software development that, in particular, examines how:

- agent-oriented analysis combined with ISM construction assists program comprehension and the understanding of requirements (Sun et al., 1998);
- software modules can be extracted from ISMs and, if necessary, optimized by translation into conventional procedural programs (Allderidge et al., 1998).
- definitive scripts provide a powerful means of data integration that can be used in particular to express the way in which low-level redefinition can entail high-level change. Data conversion agents that are empirically tuned to particular patterns of synchronization can serve as databrokers (Beynon,2000).

In the long term, Empirical Modelling can potentially support the software engineering task in finding the appropriate technology for integration. This involves the identification of new concepts and abstractions for the presentation and processing of data, files, directories at a system level. An ongoing research along this line is initiated by Cartwright [Car99] in developing the Java Maintainer[3] (JAM) application interface (API) with the aim of porting EM principles of agency and dependency to the system and objects level.

[3] JaM provides a generic dependency maintenance system for use in object-oriented programming. It also provides facilities for organising the information in a dependency structure into virtual directories and manages user and group permissions to access and modify the data and dependencies. JaM's dependency maintenance can be considered as a generic spreadsheet in which the *cells* of the

(2) The EM – VR merge

Chapter 6 suggests future prospects for the EM - VR merge in developing virtual environment for financial trading. EM technology motivates a new perspective on the design, application and use of VR technology in modelling a social context, such as the financial trading one. Further research on the prospect of the EM-VR merge is strongly advised. The aim is to identify a suitable computational framework for using VR technology in developing virtual environment for financial trading, modelling the financial trading process, supporting the financial decision making activity, and designing interfaces for virtual environments for financial trading. Qualitative and quantitative performance metrics[4] are needed to assess the distinctive qualities of model building within an EM-VR framework.

(3) Evaluation of the contribution of DEM to a wider computer-based support for the financial engineering activity:

The use of DEM in re-engineering the spreadsheet interest rate model introduced in chapter 7 suggests a potential for DEM to contribute to a wider computer-based support for the financial engineering activity that goes beyond the implementation of the mathematics of the financial model to provide support for group learning, group decision making and shared modelling. This is mainly attributed to the role of DEM in de-centralizing the spreadsheet modelling activity and extending its openness at the network level. The decentralization of the spreadsheet modelling activity involves the accommodation of various participants with diverse styles and privileges of action. The openness of the spreadsheet modelling activity at the network level involves: first the support of the distributed spreadsheet modelling with views that metaphorically represent the working space of each participant; and second the use of open ended communication of definitive script that extends endlessly the interaction of participants beyond features provided by their views. This communication does not need to follow any preconceived pattern or sequence.

It is essential for future EM research to develop ***conceptual*** and ***practical*** approaches to assess its contribution to widening the computer-based support of the financial engineering activity, and its added value in decentralizing the spreadsheet modelling activity and extending its openness at the network level to achieve a greater support for group learning among a network of financial academics.

---

spreadsheet are instances of any class chosen by a developer that implement an interface of the API, not just conventional types of dates, strings and numbers.

[4] [MBG01] suggests cognitive dimensions [Gre00] to assess the potential benefits of VR modelling in a social context.

- At a ***conceptual level***, means for such assessment can be developed through further research on the use of Interactive Situation Models for cognitive aspects of user-artefact interaction [BRWW01] in a financial engineering context.
- At a ***practical level***, the assessment may involve the recording of *quantitative* and *qualitative measurements*. *The quantitative measurements* aim at assessing an improvement at the *operational level* including: ease of use, accessibility, greater interaction and engagement of the human with the computational activity, and enhanced management of the results of the computational activity. The *qualitative measurement* aims at assessing an improvement at the *cognitive level* such as the achievement of a learning objective.

  - *Quantitative measurements* can be recorded experimentally. This may involve (in case of the use of the spreadsheet interest rate models by academics) splitting a class into two groups: the first group interacts with the spreadsheet interest rate model; while the second group interacts with the re-engineered DEM version of this model. Recording techniques may include video recording or relying on human observers who might observe the details of the interaction of the students and the teacher with both versions of the model. Details of interaction, recorded in the course of observation, may include the frequency of changing the parameters of the model, the communication among students and teachers of various changes to parameters of the model, patterns of teacher / student interaction, the control of the teacher over the modelling activity undertaken by the student (dictating its pattern when necessary), etc..

  - *Qualitative measurements* aim at assessing a learning objective and can be recorded by inviting the participants in the modelling activity (students in each of the two groups formed as described above) to write an essay or fill-in a questionnaire that reflects their understanding of the financial concepts conveyed by the model. Evidence of better meeting a learning objective emerges through comparing the median/average grade of essays / questionnaires filled-in by students of each group.

(4) EM for data intensive applications

The thesis suggests that prospects for the EM technology for data intensive application can be enhanced through further research on data referencing, processing, and interpretation. This equips EM technology with means for visual exploration of large data sets that can support qualitative analysis of data intensive application domains such as finance. Such a qualitative

analysis provides value-added advantage over quantitative analysis by supporting experiential knowledge construction, human insight, and amethodological exploration.

(5) Support of different modes of interaction in interpersonal communication in domain specific activities

The thesis suggests complex modes of interaction in computer mediated interpersonal interaction in the course of the financial engineering activity. There is sign for further research on providing a suitable framework supporting diverse forms of interpersonal interaction with various modes of agency and roles for participants in domain specific group social activities. This would extend the research undertaken in [Son93] and [Sun99].

This thesis is an initial contribution to applications and prospects for EM technology in finance. Given its broad covering of the literature in the finance and computer science domain, its overview of almost every aspect of EM technology, its tackling of major problems pertaining to the different divisions of finance, this thesis provides a solid background for further research of the application of EM technology in finance. Pursuing the case studies considered is highly advocated as they are typical case studies encountered in the field of finance.

Bibliography

[ABCY94] V.D. Adzhiev, W.M. Beynon, A.J. Cartwright, Y.P. Yung, A Computational Model for Multi-agent Interaction in Concurrent Engineering , In *Proc. CEEDA'94*, Bournemouth University, 1994, pp. 227-232.

[ABCY94-2] V.D. Adzhiev, W. M. Beynon, A. J. Cartwright, Y. P. Yung, An Agent-oriented Framework for Concurrent Engineering, In *Proc. IEE Colloquium: Issues of Cooperative Working in Concurrent Engineering*, October 1994, Digest 1994/177, 9/1-9/4.

[ABCY98] J. Allderidge, W. M. Beynon, R. Cartwright, and Y.P.Yung, Enabling Technologies for Empirical Modelling in Graphics, In *Proc. Eurographics UK, 16th Annual Conference*, 199-213, 1998.

[Abh00-1] A. Abhyankar, Transaction Costs, MSc. Lecture notes of Research Techniques in Finance, Warwick Business School, Spring 2000.

[Abh00-2] A. Abhyankar, Market Microstructure I, MSc. Lecture notes of Stock Market Theory, Warwick Business School, Spring 2000.

[Abh00-3] A. Abhyankar, Market Microstructure II., MSc. Lecture notes of Stock Market Theory, Warwick Business School, Spring 2000.

[Abh00-4] A. Abhyankar, Methodological and Statistical Issues I, MSc. Lecture notes of Stock Market Theory, Warwick Business School, Spring 2000.

[Abh00-5] A. Abhyankar, The Cross-Section of Common Stock Expected Returns II, MSc. Lecture notes of Stock Market Theory, Warwick Business School, Spring 2000.

[Abh00-6] A. Abhyankar, Portfolio Performance Evaluation, MSc. Lecture notes of Stock Market Theory, Warwick Business School, Spring 2000.

[Abh00-7] A. Abhyankar, Arbitrage Pricing Theory, MSc. Lecture notes of Research Techniques in Finance, Warwick Business School, Spring 2000.

[ABR99] V. Adzhiev, W.M.Beynon, and A. Rykhlinski, Empirical Modelling of Multi-Agent Systems with Inherent Concurrency, Department of Computer Software, The University of Aizu, Japan, June 26, 1999.

[AC98] C. Avgerou and A. Cornford, *Developing information systems: concepts, issues and practice*, 2nd edition, Macmillan Press, 1998.

[AF95] D.E. Avison, G. Fitzgerald, *Information System Development: methodologies, techniques and tools*, 2nd edition, McGraw-Hill, 1995.

[AH01] A. Abhyankar, K. Ho, Is long-horizon abnormal performance after seasoned equity offering illusory? New evidence from the UK, *Accepted Paper: Eastern Finance Association Annual Meeting, Baltimore*, April 2002

[Amb02] S. W. Ambler, *Agile Modelling: Effective Practices for Extreme Programming and the Unified Process*, John Wiley & Sons, 2002.

[AMc98] R. Alexander, M. McCann, Balanced Scorecard Measurement, In *Proc. BPR Europe 98 Conference*, 1998, London.

[And94] S.F. Andriole, Fast, cheap requirements: prototype, or else. *IEEE Software*, 11(2):85-87, 1994.

[ANM97] A. L. Ames, D. R. Nadeau, J.L. Moreland, *VRML 2.0 Sourcebook*, John Wiley & Sons, Inc, 1997.

[Arn98] G. Arnold, *Corporate Financial Management*, Financial Times, Pitman Publishing, London, 1998.

[Aus00] G. Austin, Settling Across Border – Developing Co-Operation between banks, brokers and investment managers, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[BACY94] W.M. Beynon, V. D. Adzhiev, A.J. Cartwright, Y. P. Yung, A Computational Model for Multiagent Interaction in Concurrent Engineering, In *Proc. CEEDA'94*, Bournemouth Univ., 1994, 227-232.

[BACY94a] W. M. Beynon, V. D. Adzhiev, A. J. Cartwright, and Y. P. Yung., An agent-oriented framework for concurrent engineering. In *Proc. IEE Colloquium: Issues of co-operative working in concurrent engineering*, Digest 1994/177, pages 9/1-9/4, October 1994.

[BBY92] W.M.Beynon, I.Bridge and Y.P.Yung, Agent-oriented Modelling for a Vehicle Cruise Controller, In *Proc. ESDA Conf.*, ASME PD-Vol. 47-4, 159-165, 1992.

[BC95] W.M.Beynon and R.I.Cartwright, Empirical Modelling Principles for Cognitive Artefacts, In *Proc. IEE Colloquium: Design Systems with Users in Mind: The Role of Cognitive Artefacts*, Digest No. 95/231, 8/1-8/8, December 1995.

[BCGH99] P. Bocij, D. Chaffey, A. Greasley, S. Hickie, *Business Technology Information Development Systems and Management*, Financial Times, Pitman Publishing, 1999.

[BCSW99] W.M. Beynon, R. Cartwright, P. H. Sun, A. Ward, Interactive situation models for information systems development, In *Proc. 5th International Conference on Information Systems, Analysis, and Synthesis*, pp.9-16,1999.

[BCY95] W.M. Beynon, A. Cartwright and Y.P. Yung, Databases from an Agent-oriented Perspective, Research report CS-RR-278, Department of Computer Science University of Warwick, January 1995.

[BD93] C. Britton, J. Doake, *Software System Development: A gentle introduction*, McGraw-Hill International Series in Software Engineering, 1993.

[BD93] A. Burns, G. Davies, *Concurrent Programming*, Addison-Wesley, 1993.

[Bec00] K. Beck, *eXtreme Programming Explained*, Addison Wesley, 2000.

[Ben91] M. Benedikt, Cyberspace: Some proposals, *In M. Benedikt (ED), Cyberspace: First steps*, pp 273-302, MIT Press, 1991.

[BEN98] J. Bartoo, T. Elliot, V. Naik-Iyer, Introducing E-Business Intelligence, Business Objects, *Technical white paper*, 1998.

[Bey01] M. Beynon, Liberating the Computer Arts, In *Proc. DALI'2001. First International Conference on Digital and Academic Liberty of Information*, University of Aizu, Japan, March 2001 (invited paper)

[Bey83] W.M. Beynon, A Definition of the ARCA Notation, Research Report CS-RR-054, Department of Computer Science, University of Warwick, Coventry, UK (September 1, 1983).

[Bey94] W.M.Beynon, Agent-oriented Modelling and the Explanation of Behaviour, Invited Paper in *Proc. International W/S "hape Modeling Parallelism, Interactivity and Applications*, Dept. of Computer Software, TR 94-1-040, Univ. Aizu, Japan, 54-63, Sept. 1994.

[Bey97] W.M. Beynon, Empirical Modelling for Educational Technology, In *Proc. Cognitive Technology '97*, University of Aizu, Japan, IEEE, 54-68, 1997.

[Bey98] W.M. Beynon, Modelling State in Mind and Machine, Research Report CS-RR-337, Department of Computer Science, University of Warwick, Coventry, UK (February 25, 1998).

[Bey99] W.M. Beynon, Empirical Modelling and the Foundations of Artificial Intelligence, In *Proc. CMAA'98*, Springer Lecture Notes in AI, 1999.

[BG96] R. Buck-Emden and J.Galimow, *SAPR/3 system: a client server technology*, Addison-Wesley, 1996.

[BGC98] E. de Bodt, Ph Gregoire, M. Cottrell, Projection of Long-term Interest Rates with Maps, in *Visual Exploration in Finance with Self-Organizing Maps*, Springer 1998.

[BGV97] M. Bouzeghoub, G. Gardarin, P. Valduriez, *Object Technology*, International Thomson Engineering Press, Software Engineering, 1997.

[BJ94] W.M.Beynon and M.S.Joy , Computer Programming for Noughts-and-Crosses: New Frontiers, In *Proc. PPIG'94, Open University*, 27-37, January 1994.

[BJS72] F. Black, M.C. Jensen, M. Scholes, The Capital Asset Pricing Model: Some Empirical Tests, in *Studies in the Theory of Capital Markets*, Micheal C. Jensen, ed. (New York: Praeger Publishers, 1972).

[BKM96] Z. Bodie, A. Kane, and A.J. Marcus, *Investments*, McGraw-Hill, 1996

[Bla01] M. R. Blaha, *A manager's guide to database technology: building and purchasing better applications*, Prentice Hall, 2001.

[Bla72] F. Black, Capital Market Equilibrium with Restricted Borrowing, *Journal of*

Business, 1972, 45, July, pp.444-454.

[BM00] W. M. Beynon, S. Maad, Integrated Environments for Virtual Collaboration: an Empirical Modelling Perspective, In *Proc. The Fifth World Conference On Integrated Design & Process Technology, Incorporating IEEE International Conference on Systems Integration*, Texas, USA, 2000.

[BM99] W. M. Beynon, S. Maad, Empirical Modelling of Real Life Financial Systems: The need for Integration of Enabling Tools and Technologies, In *Proc. The Fourth World Conference On Integrated Design & Process Technology, Incorporating IEEE International Conference on Systems Integration*, Texas, USA, 1999.

[BNRSYY89] W.M. Beynon, M.T. Norris, S.B.Russ, M.D. Slade, Y.P. Yung, Y.W. Yung, Software Construction Using Definitions: an Illustrative Example, Research Report CS-RR-147, Department of Computer Science, University of Warwick, Coventry, UK (September 1, 1989)

[Boe88] B.W. Boehm, A spiral model of software development and enhancement, *IEEE Computer*, 21(5):61-72, 1988

[Boo86] G. Booch, Object-Oriented Development, *IEEE Trans. S.E.* 12(2), 1986, 285-292.

[Boo91] G. Booch, *Object Oriented Design with Applications*, Benjamin Cummings, Redwood, CA, 1991.

[Boo94] G. Booch, *Object Oriented analysis and design with application*, Benjamic/Cummings, 1994.

[Bos89] R.P. Bostrom, Successful application of communication techniques to improve the systems development process. *Journal of Information and Management*, 16:279-295,1989

[Bou96] T. Boutell, *CGI Programming in C & Perl: Featuring C and Perl 5 Source Code*, Addison-Wesley, 1996.

[Bow88] D.S. Bowers, *From data to databases*, Champan & Hall, 1988. pp.1-19

[Bra00] M. Bradbury, Increasing the efficiency of cross border securities trading, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September

2000, London.

[Bro83] R. Brook, Towards a theory of comprehension of computer programs, *Int'l J. Man-Machine Studies*, Vol.18, 1983, 543-554.

[Bro87] F.P.Brooks, Jr., No silver bullet: *essence and accidents of software engineering*, *IEEE Computer*, 20(4):10-19, 1987.

[Bro91] R. A. Brooks. Intelligence without representation. *Artificial Intelligence,* 47:139-159,1991

[Bro91-2] R. A. Brooks. Intelligence without reason. In *Proc. IJCAI-91,* pages 569-595, 1991.

[BRo93] M. Bouzerghoub, A. Rochfeld, *De Merise a OOM*, Hermes, Paris, 1993

[BRo94] M. Bouzerghoub, A. Rochfeld, *Objets du C++ a Merise*, Eyrolles, Paris, 1994.

[BRO95] F.P. Brooks, The Mythical Man-Month Revisited: *Essays on Software Engineering*, Addison-Wesley, 1995.

[BRR00-1] W.M.Beynon, S. Rasmequan and S. Russ, An Experience-Based Approach to Decision Support Systems, *Position paper for IFIP Working Group 8.3, Working Conference on Decision Support Through Knowledge Management*, Stockholm, Sweden, July 9-11, 2000.

[BRR00-2] W.M. Beynon, S. Rasmequan and S. Russ, The Use of Interactive Situation Models for the Development of Business Solutions, In *Proc. workshop on Perspective in Business Informatics Research (BIR-2000)*, Rostock, Germany, March 31- April 1, 2000.

[BRR00-3] W.M. Beynon, S. Rasmequan and S. Russ, A New Paradigm for Computer-Based Decision Support, *the Journal of Decision Support Systems*, Special Issue, Autumn 2000.

[BRS99] W.M.Beynon, J. Rungrattanaubol and J. Sinclair, Formal Specification from an Observation-Oriented Perspective, *15th British Colloquium in Theoretical Computer Science*, Keele University, April 1999.

[BRSW98] W.M. Beynon, J. Rungrattanaubol, P. H. Sun, A. E. M. Wright, Explanatory Models for Open-ended Human-Computer Interaction, Research report

CSRR-346, Warwick Univ., 1998

[BRU94] W. M. Beynon, S. Russ, Empirical Modelling of Requirements, Research Report CSRR-277, Department of Computer Science, University of Warwick, UK, 1994.

[BRWW01] W.M.Beynon, C. Roe, A. Ward, A. Wong, Interactive Situation Models for cognitive aspects of user-artefact interaction, In *Proc. of Cognitive Technology Conference (CT2001)*, Warwick University, UK, August 2001.

[Bry82] J. W. Bryant, *Financial Modelling in Corporate Management*, 1982, pp.3-55, p345, pp 427-444.

[BS98] W.M. Beynon, P.H. Sun, Interactive Situation Models for Program Comprehension, Research report CSRR-352, Department of Computer Science, University of Warwick, Coventry CV47AL UK, 1998.

[BS99] W.M.Beynon and P-H. Sun, Computer-mediated communication: a Distributed Empirical Modelling perspective, In *Proc. CT'99*, San Francisco, August 1999.

[BWMRR00] W.M.Beynon, A. Ward, S. Maad, A. Wong, S. Rasmequan, and S. Russ, The Temposcope: a Computer Instrument for the Idealist Timetabler, In *Proc. the 3rd international conference on the practice and Theory of Automated Timetabling,Constance*, Germany, August 16-18, 2000.

[CABDGHJ94] D. Coleman, P.Arnold, S.Bodoff, C. Dollin, H. Gilchrist, F. Hayes, P. Jeremes, *Object Oriented Development: The fusion Method*, Prentice Hall, Englewood Ciffs, NJ, 1994.

[Car00] R.I. Cartwright, Empirical Worlds – Internet-based applications for shape modelling, *EUROGRAPHICS'2000*, Volume 19, (2000), Number3.

[Car99] R. Cartwright, *Geometric Aspects of Empirical Modelling: Issues in Design and Implementation*, Phd Thesis, Department of Computer Science, University of Warwick, 1999.

[CB91] D. E. Carter and B. S. Baker, *Concurrent Engineering: The Product Development Environment for the 1990s*. Addison-Wesley, 1991.

[CCH80] G.A. Champine, R.D. Coop, and R.C. Heinselman, *Distributed computer*

systems: impact on management, design and analysis, North-Holland Publishing, 1980.

[CDK94] G. Coulouris, J.Dollimore, and I. Kindberg, *Distributed system: concepts and design*, Addison-Wesley, 1994.

[CFW94] H. Chi-Cheng, B. R. Fuller, and K. Wenchi, A Method For Inferring the Effective Bid-Ask Spread From Security Returns, *Journal of Business Finance & Accounting*, 21(2), March 1994, 0306-686x, pp.243-253.

[Cha85] C. Chatfield, The Initial Examination of Data, *Journal of the Royal Statistical Society*, 148, Part3: 214-253, 1985.

[Cha98] P. S.Chan, *Support for an Integrated Approach to Program Understanding: An Application of Software Visualization*, Phd Thesis, 1998, Chapter 5.

[Che76] P.Chen, The entity relationship model – *toward a unified view of data, ACM Trans. On Data Base Systems*, 1(1):6-36, 1976.

[Chr00] K. Christopherson, Body of the New Economy, *Conspectus, E-business Solutions*, December 2001.

[Cla01] A. Clark, Natural-Born Cyborgs?, In *Proc.* The Fourth International Conference on Cognitive Technology: Instruments of Mind, August 2001, Warwick University, UK.

[Cla97] W.J. Clancey, *Situated Cognition: on human knowledge and computer representation*, Cambridge University Press, 1997.

[CLM97] J. Campbell, W. A. Lo, and A.C. MacKinlay, *The Econometrics of Financial Markets*, Princeton University Press, NJ, 1997.

[CM97] P.S. Chan, M. Munro, PUI: A Tool to Support Program Understanding, In *Proc. the IEEE 5th International Workshop on Program Comprehension*, Dearborn, Michigan, IEEE Computer Society Press .pp. 192-198, May 28-30, 1997.

[COM94] The Committee for Advanced DBMS Function, Third Generation Database Systems Manifesto, from *Readings in Database Systems*, 2nd ed., Morgan Kaufman, 1994.

[COMICAL] COMICAL, Cognitive Observation-Oriented Modelling for Interactive

Computer-aided Learning, *Workshop, hosted by Empirical Modelling Group*, Department of Computer Science, University of Warwick, UK, Summer 1999.

[Cor99] C. Corvey, Passive Asset Allocation, MSc. Lecture notes of Investment Management, Warwick Business School, Fall 1999.

[Cov00] J. Coval and T. Shumway, Is Sound Just Noise?, In *Proc. 27th Annual Meeting of European Finance Association*, August 2000, London.

[CRB00] Y-C. Chen, S. Russ and W.M. Beynon , Participative Process Modelling, In *Proc. IEEE SMC 2000 Cybernatics Evolving to Systems, Humans, Organizations and their complex interactions*, Tennessee, USA, October 8-11, 2000.

[Cro94] C. Crook, *Computers and the collaborative experience of learning*, Routledge, 1994.

[CRV99] P. Ciancarini, D. Rossi, F. Vitali, A case study in designing a document-centric coordination application over the internet, In *Proc. The Active Web Conference*, January 99, Staffordshire University, UK.

[CS90] P. Checkland, J. Scholes, *Soft system methodology in action*, John Wiley and Sons, 1990.

[CT01] Introduction, In *Proc.* The Fourth International Conference on Cognitive Technology: Instruments of Mind, August 2001, Warwick University, UK.

[CUT99] Executive Summary: Distributed Computing Architecture Advisory Service, *Cutter Consortium Executive Report*, Cutter Consortium, 1999 Executive Report, Vol.2, No.2.

[CW96] J.A. Campbell and J. Wolstencroft. Structure and significance of analogical reasoning. *Al in Medicine*, 8(2):103-118, 1996.

[CW98] D.J. Carney, K.C. Wallnau, Pro/COTS:Process Issues in Building COTS-based Systems, In *Proc. ICSEA Conference*, Vol. 2, 1998, Paris.

[CWG93] E. Carmel, R.D. Whitaker, and J.F. George, PD and joint application design: a transatlantic comparison. *Communications of the ACM*, 36(4):40-48, June 1993.

[CY90] B.J.Cox and E. Yourdon, *Object-Oriented Analysis*, Prentice-Hall, 1990.

[CYo91] T. Coad, E. Yourdon, *Object Oriented Analysis*, 2nd edition, Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ., 1991.

[Dav91] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991

[Dav93] T.H. Davenport, Process Innovation: *Re-engineering work through information technology*, Boston: Harvard Business School Press, 1996.

[Dav93-A] A.M. Davis, *Software Requirements objects, functions, and states*, Prentice-Hall International, Inc. 1993.

[Dav96] E.L. Davis, Modelling Human Interaction, 3rd Year Project Report, Department of Computer Science, University of Warwick, May 1996.

[Dav96-2] T. H. Davenport, Why re-enginering failed: The fad that forgot people, in *Fast Company*, Premier Issue, pp.70-74, 1996.

[Deb94] G.J. Deboeck, *Trading on the edge*, John Wiley, New York, 1994.

[Deb98] G. Deboeck, Investment Maps of Emerging Markets, In *Visual Exploration in Finance with Self Organizing Maps*, Springer, 1998, pp.83-105.

[Deb98-2] G. Deboeck, Picking Mutual Funds with Self-Organized Maps, In *Visual Exploration in Finance with Self-Organizing Maps*, Springer 1998.

[Den87] D.C. Dennett, *The intentional stance*, the MIT Press, 1987.

[Der87] D. McDermott. A critique of pure reason. *Comput Intell*, 3:151-160, 1987.

[Dew16] J. Dewey, *Learning Through Doing*, 1916.

[DHM93] B. Delatte, M. Heitz, J.F. Muller, *HOOD Referencee Manual 3.1*, Masson, Paris, 1993.

[DK98] G. Deboeck, T. Kohonen , *Visual Explorations in Finance with Self-Organizing Maps*, Springer Finance, 1998.

[DN66] O.J. Dahl, K. Nygaard, Simula – an Algol based simulation language. *Communications of the ACM*, 9(9), 671-8, 1966.

[Dor98] H.P.D'Ornellas, *Agent-oriented modelling for collaborative group learning*, MSc, Department of Computer Science, Warwick University, 1998.

[DPe98] Z. Dienes, J. Perner, A Theory of Implicit and Explicit Knowledge. *Behavioral and Brain Sciences* 22 (5): 1998: Cambridge University Press

[Drey79] H.L. Dreyfus, *What Computers Can't do*, Harper and Row, New York, 1979.

[Eas99] G. Eastwood, Web-enabled ERP, Computer Business Review, Special Report Series, *ERP The Next Generation*, 1999, pp 17-20.

[EASDAQ] EASDAQ, The pan-European stock market for growth and technology companies, 1998 report.

[Ebe99] J.M. Eber, What can finance learn from computer science, In *Proc. Options Conference*, Warwick University, June 1999.

[Eck98] B.E. Eckbo, D. Smith, The Conditional Performance of Insider Trades, *Journal of Finance*, 1998, 53(2), pp.467-498.

[EG95] E.J. Elton, M.J. Gruber, *Portfolio Theory and Investment Analysis*, Wiley, 1995

[EGH93] R.A. Earnshaw, M.A. Gigante, H. Jones, *Virtual Reality Systems. Academic Press*, (1993).

[Eld00] M. Elderfield, Tackling the regulatory implications of global online trading, In *Proc. of On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[Elt95] E.J. Elton, M.J. Gruber, *Portfolio Theory and Investment Analysis*, Wiley, 1995

[EM87] D. Edwards, N. Mercer, *Common Knowledge*, London: Methuen, 1987.

[EMB94] F.J. Erens, A. McKay, A., S. Bloor, Product Modelling Using Multiple Levels of Abstraction: Instances as Types, *Computers in Industry*, Vol.24, No.1, 17-28, 1994.

[ESc93] M.J. Euwe, R.V. Schuwer, Configuration of Complex Products, *Computers in Industry*, Vol.21, 1-10, 1993.

[Ess93] J. Essinger, *Managing technology in Financial Institutions*, Financial Times, Pitman Publishing, 1993.

[EXC98] *Excelon*, Product description brochure, An XML data server developed by Object Design, Inc., 1998.

[Fam70] E. F. Fama, Efficient Capital Markets: A Review of Theory and Empirical Work, *Journal of Finance*, 1970, XXV, 25, pp.383-417.

[Fam91] E. F. Fama, Efficient Capital Markets II, *Journal of Finance*, 1991, 46, pp. 1575-1618.

[FB01] C.N. Fischer, W.M. Beynon, Empirical Modelling of Products, In *Proc. International Conference on Simulation and Multimedia in Engineering Education*, Phoenix, Arizona, January 2001, p20-26, The Society for Modelling and Simulation International, 2001, ISBN: 1-56555-224-5

[FB01-2] C.N. Fischer, W.M. Beynon, Empirical Modelling in product design, In *Proc. 16th Brazilian Congress of Mechanical Engineering*, November 2001, p218-225, ISBN 85-85769-06-6

[FF91] R.P. Frail, R.S. Freedman, Name Classification in Noisy Domains, In *Proc. The First International Conference on Artificial Intelligence Application on Wall Street*, IEEE Computer Society Press, October 9-11, 1991, New York.

[FFJR70] E. Fama, L. Fisher, M. Jensen, and R. Roll, The Adjustment of Stock Prices to New Information, *International Economics Review*, Vol. 10, 1-21, 1970.

[FG97] S. Franklin and G. Graesser, Is it an agent, or just a program?: A taxonomy for autonomous agents. *In J.P. Muller, M.J. Wooldridge, and N.R. Jennings, editors, Intelligent Agents III – Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), Lecture Notes in Artificial Intelligence, ECAI-96*, Budapest, 1997.

[Fit96] B. Fitzgerald, Formalized System development methodologies: a critical perspective. *Information System Journal*, 6:3-23, 1996.

[FK95] K. Feldman, J. Kingdon, Neural Network and some applications in finance, *Journal of Applied Mathematical Finance*, 1995, 1, 1.

[Flo87] C. Floyd, Outline of a paradigm change in software engineering, In *G. Bjerkness, P. Ehn, and M. Kyng (eds), Computer and democracy: a Scandinavian challenge*, pp.191-212, Avebury, Aldershot, 1987.

[Flo95] C. Floyd, Theory and practice of software development. *In P.D. Mosses, M. Nielsen, and M.I. Schwartzbach (eds.), TAPSOFT'95:theory and practice of*

software development, *Lecture Notes in Computer Science*, 915:25-41, 1995.

[FRKBCJ00] K. Fernandes, V. Raja, J. Keast, W.M. Beynon, P.S.Chan, M. Joy, Business and IT perspectives on AMORE: a methodology using Object-Orientation in Re-engineering Enterprises, In *Systems Engineering for Business Process Change: New Directions*, (ed P. Henderson) Springer-Verlag, December 2001, ISBN 1-85233-399-5.

[FS99] T. Flanagan, E. Safdie, *A practical guide to achieving enterprise data quality*, The technology guide series, techguide.com, Trillium Software System, 1999.

[Gar67] H. Garfinkel, *Studies in ethnomethodology*, Prentice-Hall, 1967.

[GC00] Garbaya, S. and Coiffet, P., Generating Operation Time from Virtual Assembly Environment, In *Proc. of the 7th UK VR-SIG Conference*, 19th September, 2000, University of Strathclyde, Glasgow, Scotland

[GHUW94] B.C. Glasson, I.T. Hawryszkiewycz, B.A. Underwood, R.A. Weber, *Business Process Re-engineering: Information Systems Opportunities and Challenges*, Amsterdam: Elsevier Science, 1994.

[Gib92] R. Gibbons, *A Primer in Game Theory*, Harverster Wheatsheaf, 1992.

[Gog94] J.A. Goguen, Requirements engineering as the reconciliation of technical and social issues. In *M. Jirotka and J. Goguen (eds), Requirements Engineering: Social and Technical Issues*, pp.165-200, Academic, 1994.

[Gol88] R. K. E. Goldman, *Computerized Trading Strrategies: How to Program for the Stock and Futures Markets*, John Wiley & Sons, Inc., 1988.

[Goo90] D. Gooding, *Experiment and the making of meaning*, Kluwer Academic Publishers, 1990.

[GP92] T.R.G. Green, M. Petre, When Visual Programs are harder to read than textual programs, In *G.C. van der Veer, Jauber M.J. Bagnarola S., Antavolits (Eds.) Human-Computer Interaction: Tasks and Organisation, Proc. ECCE-6*, 1992.

[GRA94] I. Graham, O*ject-Oriented Methods*. Addison-Wesley. Harlow, 1994, Second edition.

[GRA98] S. McGrath, *XML By Example Building E-commerce Applications*, The

Charles F.Goldfarb Series on Open Information Management, 1998.

[Gre00] T.R.G. Green, Instructions and Descriptions: some cognitive aspects of programming and similar activities, In *Proc. of working Conference on Advanced Visual Interfaces (AVI 2000)*. New York: ACM Press, pp21-28, 2000.

[Gre00-2] R. Greifield, Integration and internationalisation of online order flow into traditional markets and ECN's, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[Gri95] M. Grinblatt, S. Titman, Performance Evaluation, Chapter 19 in *Jarrow, Robert A, Vojisalav Maksimovic and William T Ziemba (eds.)*, 1995 Finance, Handbooks in Operations Research and Management Science, North Holland.

[Gur99] A. Gurton, Business Guide To Supply Chain, *Information Age*, April 99, pp.34-38.

[Hal98] M. I. Hale. *The mind: Its origin, evolution, structure and functioning*. Hale-van Ruth, Pittsburgh, 1989.

[Han99] R. Hanbury, Back-end integration, *Internet Business Magazine*, July 99, Issue 30, pp 46-48.

[Har88] L.E. Harris, Estimation of True Stock Price Variance and Bid-Ask Spread from Discrete Observations, Working Paper, University of southern California, June 1988.

[Har98] L. Harris, *Trading and Exchanges*, Draft textbook: December 4, 1998, University of Southern California, Marshall School of Business.

[Har99] L. Harasim, A Framework For Online Learning: The Virtual-U, *IEEE Computer Society*, Computer, September 1999, pp. 45-49

[Hay00] A. Haynes, Access on-line order-matching systems to trade anonymously and at low cost, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[HBR97] D.E. Hartman, T. Brock, Extracts From A Virtual Poem: Extending Classroom Reality?, *Computer Graphics*, August 1997.

[HBS73] C.E. Hewitt, P. Bishop, and R. Steiger, A universal modular actor formalism for artificial intelligence, In *Proc. International Joint Conference on AI*, Stanford, California, 235-45, 1973.

[Hee98] R. Heeks, Why Information Systems Succeed Or Fail: Conception-Reality Gaps, In *Proc. BIT98 Conference*, 1998, Manchester.

[HFK94] C. Hsia, B. R. Fuller, and G. W. Kao, A Modified Method for Inferring the Effective Bid-Ask Spread from Security Returns, *Journal of Business Finance and Accounting* 21(2), March 1994, pp. 243-254.

[HKL95] R. Hirschheim, H. K. Klein, K. Lyytinen, *Information systems development and data modelling: conceptual and philosophical foundations*. Cambridge University Press, 1995.

[HKN91] R. Hirschheim, H. K. Klein, M. Newman, *Information systems development as social action: theoretical perspective and practice*. OMEGA, 19(6):587-608, 1991.

[Ho00] K. Y. Ho, *Essays on Long-Horizon Event Studies: An Exploration of Asset Pricing and Econometrics Issues*, Mphil/Phd Upgrading Report, Warwick Business School, 2000.

[Ho00-2] K. Y. Ho, *Essays on Long-Horizon Event Studies: An Exploration of Asset Pricing and Econometrics Issues*, departmental presentation, Warwick Business School, 2000.

[HO93] W. Harrison, H. Ossher, Subject-oriented Programming (A Critique of Pure Objects), In *Proc. OOPSLA'93*, 1993, pp. 411-428.

[Hon99] H.S. Hong, *Empirical Modelling For Financial Systems*, MSc thesis. Department of Computer Science, University of Warwick, 1999.

[HSH98] R. Hoque, T. Sharma, *Programming Web Components*, McGraw-Hill, 1998.

[Hum98] Hummingbird Communications LTD., Web-based Business Intelligence and The Low Cost of Ownership Imperative, A white Paper, 1998.

[Hut95] E. Hutchins, *Cognition in the Wild*, MIT Press, 1995.

[HW91] P.J. Hayes, S.P. Weinstein, Adding Value to Financial News by Computer, In *Proc. The First International Conference on Artificial Intelligence*

Application on Wall Street, IEEE Computer Society Press, October 9-11, 1991, New York.

[HWE96] J. Hartman, J. Wernecke, *The VRML 2.0 Hanbook – Building Moving Worlds on the Web*, Silicon Graphics Inc, 1996.

[IBM98] IBM's Visual Warehouse and Data Cleansing and Reconciliation, Technical Brief, IBM & Vality Technology Incorporated, 1998.

[Inv98] Investment Company Institute, *A Guide to Mutual Funds*, 1998, USA.

[Jam01] /dcs/acad/wmb/public/projects/notations/JaM2API/DOCS/JaM2

[Jam07] W. James. What Pragmatism Means. Lecture 2 in *Pragmatism: A new name for some old ways of thinking*. New York: Longman Green and Co (1907): 17-32.

[JCJO92] I. Jacobson, M. Christeron, P. Jonson, G. Overgaard, *Object Oriented Software Engineering: a use case driven approach*, Addison-Wesley, 1992

[JEJ95] I. Jacobson, M. Ericsson, A. Jacobson, *The Object Advantage: Business Process Reengineering with Object Technology*, Addison-Wesley, 1995.

[Jen00] R. Jennings, R. Battalio, and B. Hatch, Dimension of Best Execution For Market Orders: Assessing Differences Between the NYSE and Nasdaq's Third Market, *In Proc. 27th Annual Meeting of European Finance Association*, August 2000, London.

[Jen97] Jennings N.R., Wooldridge M., *Applications of Intelligent Agents*, Queen Mary & Westfield College, University of London, 1997.

[JES00] S.P. Jones, J. Eber, J. Seward, Compositional description and valuation of financial contracts, January 2000.

[JHA98] SL P. Jones, RJM Hughes, L. Augustsson, D. Barton, B Boutel, W. Burton, J Fasel, K Hammond, R Hinze, P Hudak, T Johnsson, MP Jones, J. Launchbury, E Meijer, J Peterson, A Reid, C Runciman, and PL Wadler, Report on the programming language Haskell 98, Technical report, February 1998.

[JW00] J. James, N. Webber, *Interest Rate Modelling*, John Wiley, London, 2000.

[Kab98] M. J. Kabir, *Apache Server Bible*, IDG Books Worldwide, Inc, 1998.

[Kai99] H. Kaindl, Difficulties in the transition from OO Analysis to Design, In *IEEE Software*, Sseptember/October 1999, pp.94-102.

[Kar98] I. Karatzas, S. E. Shreve, *Methods of Mathematical Finance*, Springer publications 1998.

[Kat75] H. Jr. Katzan, *Introduction to Computer Science*, Mason/Charter Publishers, Inc. 1975.

[Kic91] R. C. Kick, The Strategic Significance of Expert Systems, In *Proc. The First International Conference on Artificial Intelligence Application on Wall Street*, IEEE Computer Society Press, October 9-11, 1991, New York.

[Kil00] R. Kilsby, Launching a Pan European trading system, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[Kin97] J. Kingdon, *Intelligent Systems and Financial Forecasting*, Springer, 1997.

[Kir91] D. Kirsh. Foundations of Al: the big issues. *Artificial Intelligence*, 47:3-30,1991

[KJ98] M. Knapik and J.Johnson, *Developing intelligent agents for distributed systems*, McGraw-Hill, 1998.

[KMB00] P. Konana, N. M. Menon, and S. Balasubramanian, The implications of online investing, *Communication of the ACM*, Volume 43, Number 1, January 2000, pp.35-41

[Kol96] W. R. Kolb, J. R. Rodriguez, *Financial Management*, Blackwell Publishers, Inc., 1996.

[Kow79] R. Kowalski, *Logic for problem solving*, North Holland, 1979.

[Kru83] M. Krueger, *Artificial Reality*, Reading, Mass : Addison-Wesley, 1983, p.xi.

[KS86] H.F. Korth, A. Silberschatz, *Database System Concepts*, McGraw-Hill International Editions, 1986.

[Kyn91] M. Kyng, Designing for cooperation: cooperating in design. *Communications of the ACM*, 34(12):65-73,1991.

[Lan00] L. J. Langton, The Future of Trading Online – Competition and Consolidation in the Online Securities Market Association, In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[Lan99] L. J. Langton, The impact of the electronic marketplace: The transformation of securities trading, In *Proc. On-Line Securities Trading 1999*, AIC Worldwide Ltd, September 1999, London.

[LBBRS97] J. Landauer, R. Blach, M. Bues, A. Rosch, A. Simon, Toward Next Generation Virtual Reality Systems, VISLab, Fraunhofer Institute for Industrial Engineering, Germany, *IEEE Multimedia Computing and Systems*, 1997.

[LF91] D.B. Lenat and E.A. Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47(1):185-250, 1991

[LI95] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proc. 1st Inter. Conf. on Multi-Agent Systems*, pages 254-260. MIT Press, 1995

[LIA98] B. Liautaud, Shared Knowledge, *Enterprise Business Intelligence quarterly Briefing from Business Objects*, Autumn 1998.

[Lin65] J. Lintner, Security Prices, Risk and Maximal Gain from Diversification, *Journal of Finance*, 1965, 20, pp.587-615.

[Loc95] J. Locke, Applying Virtual Reality: Making the unreal seem real, *IEEE potentials*, 0278-6648/95, 1995.

[LON98] K. Long, Process Modelling Tools, Demo's, and Evaluations, In *Proc. BPR Europe98*, 1998, London.

[Lov83] M. C. Lovell, Data Mining, *The Review of Economics and Statistics Journal*, February 1983, Vol. LXV, Number1.

[LZ75] B.H. Liskov, S.N. Zilles, Specification techniques for data abstractions, *IEEE Transactions on Software Engineering*, SE-1, N 1, 7-18, 1975.

[Maa02-Web] S. C. Maad, *An Empirical Modelling Approach to Software System Development in Finance: Applications and Prospects*, CDROM of Phd Thesis, Department of Computer Science, University of Warwick, 2002.

[Maa99] S. Maad, An Empirical Modelling approach to The Integration of E-commerce and ERP applications, *Research report*, Department of Computer Science, University of Warwick , UK, 1999.

[Mac89] B.J. MacLennan, *Functional Programming*, Addison Wesley, 1989.

[MAG01] Magnet Communications Survey, Leading Banks Say Number Of Businesses Using Internet Banking Is Increasing, In *Proc. Magnet User Conference Reveals Future Of Internet Business Banking Atlanta, Georgia* -- Atlanta, GA, July 26, 2001 – Magnet Communications, Inc. (www.magnetbanking.com).

[Mal99] B.G. Malkiel, *A Random Walk Down Wall Street*, W.W. Norton & Company, 1999.

[Mar00] A. Martin, The evolution of an Information Systems Development simulation, *journal of Simulation & Gaming*, 2000.

[Mar59] H. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*, John Wiley, New York, 1959.

[Mar98] R. T. Marshak, Open Text Livelink Intranet: Expanding Document Management to Support Collaborative Team Projects, Patricia Seybold Group's Workgroup Computing Report Vol. 21, No. 7, 1998.

[Mas89] G. Masini, *Les language a objets*, Intereditions, Paris, 1989.

[Mat01] R. Mattu, *The personal touch in understanding customer relationship management*, FT, winter 2001.

[Mat80] H.R. Maturana, Biology of cognition. In *H.R Maturana and F.J. Varela (eds), Autopoiesis and cognition: the realization of living*, pp.2-62, Reidel, 1980.

[Mat99] R. Mattison, *Web Warehousing and Knowledge Management*, McGraw-Hill, 1999.

[MBG01] S. Maad, W. M. Beynon, S. Garbaya, Realising Virtual Trading: What Price Virtual Reality?, *Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents and Virtual Reality, M.J. Smith, G. Salvendy, D. Harris, R.J. Koubek (editors)*, Lawrence Erlbaum Associates, Mahwah, N.J., pp. 1007 - 1011, ISBN 0-8058-3607-1, Volume 1

[MCB99] A. Martin, H. Christiaens, S. Brown, A customizable, computer-based interactive information systems development project simulation, In *Saunders D and Severn J (Eds.)(1999) The International Simulation & Gaming*

Research YearBook, Vol. 7, London: Kogan Page, pp 229-241.

[McG98] S. McGrath, *XML By Example Building E-commerce Applications*, The Charles F.Goldfarb Series on Open Information Management, 1998.

[Mec91] Meckler, *Virtual Reality 91*, Conference Forum, London, 5 June 1991.

[Mil99] K. Milne, How firms can optimize performance and enhance their effectiveness by implementing straight through processing, In *Proc. On-Line Securities Trading 1999*, AIC Worldwide Ltd, September 1999, London.

[MRE98] A. Moissis, C. Read, Business Intelligence and the Web: The Convergence of Two Technologies That Will Change The Way We Make Decisions, Technical White Paper, Business Objects, 1998.

[MRT76] H. Mintzberg, D. Raisinghani, A. Theoret, The structure of "unstructured" decision processes, *Administrative Science Quarterly* Vol. 21, June 1976, pp.246-275.

[MT94] C. Machover, S. E. Tice, Virtual Reality, *IEEE Computer Graphics & Applications*, January 1994.

[MTS99] C. J.P. Moschovitis, H. Poole, T. Schuyler, T. M. Senft, *History of the Internet – A Chronology, 1843 to the Present*, ABC-CLIO, The Moschovitis Group, Inc., 1999.

[Mum95] E. Mumford, *Effective system design and requirement analysis: the ETHICS approach.* Macmillan Press Ltd, 1995

[NAR93] B. A. Nardi, *A small matter of programming: perspectives on end-user computing*, MIT Press, 1993.

[NASD99] The National Association of Securities Dealers, *Glossary of Investing Terms: Your Guide To Making Safe, Smart, and Rewarding Investments*. NASD Publication, 1999.

[Nes97] P.E.Ness, *Creative Software development – An Empirical Modelling framework*, PhD thesis. Department of Computer Science, University of Warwick, 1997.

[Nor83] D. A. Norman. Some observations on mental models. In *D. Gentner and A.L.*

Stevens (eds.), Mental Models, pp.7-14, Lawrence Erlbaum Associate Inc, 1983.

[NS72] A. Newell, H. A. Simon, *Human Problem Solving*, Englewood Cliffs, NJ: Prentice Hall, 1972.

[NS76] A. Newell, H. A. Simon, Computer Science as empirical inquiry: Symbols and search, *Communications of the ACM* 19(3):113-126, 1976.

[Ode00] T. Odean and B. Barber, Online Investors: Do The slow Die First?, In *Proc. 27th Annual Meeting of European Finance Association*, August 2000, London.

[PAP71] S.Papert - *Teaching children thinking* - (1971).

[PAP80] S.Papert , *Mindstorms : Children, computers and powerful ideas* - (1980).

[PAP93] S.Papert, *The children's machine : Rethinking school in the age of the computer*, 1993.

[PAP98] S.Papert, *Technology in schools - To support the system or render it obsolete*, 1998.

[Pen87] N. Pennington, Comprehension strategies in programming, *Empirical Studies of Programmers: Second Workshop*, New Jersey, Alex Publishing Co. 1987, 100-113.

[Per99] H. Perks, Online Securities Trading, In *Proc. On-Line Securities Trading 1999*, AIC Worldwide Ltd, September 1999, London.

[Pie78] C. Peirce, How to Make Our Ideas Clear, In the *Popular Science Monthly*, January 1878.

[Por99] R. Porter, Integrating Databases, *Genio Seminar*, Leonard's Logic, 1999.

[Pot93] C. Potts, Software Engineering research re-visited, *IEEE Software*, 10(5):19-28, September 1993.

[Pro91] *Proceeding of* The First International Conference on Artificial Intelligence Application on Wall Street, IEEE Computer Society Press, October 9-11, 1991, New York.

[PRSBHC94] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, *Human-Computer Interaction*, Addison-Wesley, 1994.

[QUA99] N. Quacquareli, E-Business – the next generation, *Top Technical Graduate Career Guide*, Spring 1999, pp. 6-7.

[Rac90] S. Rachelle, The role of hypermedia in education: a look at the research issue, *Journal of Research on Computing in education*, 22(4), 431-441, 1990.

[Raj99] F. Rajola, Credit Risk Knowledge Management in the Banking Industry, In P*roc. the 9th Annual Conference on Business Information Technology Management: Generative futures*, November 1999, Manchester Metropolitan University, UK.

[RB00] D. Rime and G. H. Bjonnes, FX Trading … LIVE! Dealer Behavior and Trading Systems in Foreign Exchange Markets, In *Proc. 27th Annual Meeting of European Finance Association*, August 2000, London.

[RB74] J. Radford and A. Burton. *Thinking: its nature and development*. John Wiley & Sons, New York, 1974.

[RBPEL91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object Oriented Modelling and Design*, Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ., 1991.

[RCO98] B. Riley, P. Coggan, G. Oconnor, J. Eaglesham, and D. Harrison, Wealth Course, *A Financial Times Guide to Managing Your Money*, Financial Times in association with M&G, 1998.

[RCR00] S. Rasmequan, C. Roe and S. Russ, Strategic Decision Support Systems: An Experience-Based Approach, In *Proc. 18th IASTED Conference on Applied Informatics*, Innsbruck, Austria, February 14-17 2000.

[Rei92] B. Reinard, *Prototyping, an approach to evolutionary system development*, Springer-verlag, 1992.

[Rei99] P. Reichardt, Clearing and settlement online: implications for securities setllement arising from straight through processing, In *Proc. On-Line Securities Trading 1999*, AIC Worldwide, September 1999, London.

[RHHHR98] H. Robinson, P. Hall, F. Hovenden, and J. Rachel, Postmodern software development, *The Computer Journal*, 41(6):363-375, 1998.

[rhyme] http://www.rhymezone.com/

[Rid93] M. Ridley, Mathematics of markets, *Economist survey: Frontiers of Finance*, 9 October 1993.

[Riv00] R. Rivero, "The role of the intelligent vendor", In *Proc. On-Line Securities Trading 2000*, AIC Worldwide Ltd, September 2000, London.

[RJL97] M. Roussos, A. E. Johnson, J. Leigh, C. A. Vasilakis, C. R. Barnes, T.G. Mother, NICE: Combining Constructionism, Narrative and Collaboration in a Virtual Learning Environment, *Computer Graphics*, August 1997.

[RK91] E. Rich, K. Knight, *Artificial Intelligence*, Second Edition, McGraw Hill, Inc., 1991.

[RN95] S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[Roe00] C. Roe, Data Analysis through Definition Driven Interactive Exploration, Working paper, Department of Computer Science, University of Warwick, 2000.

[Roe99] C. Roe, *The EMPTY Project*, MSc thesis, Department of Computer Science, University of Warwick, UK, 1999.

[Rog83] M.F. Rogers, *Sociology, ethnomethodology, and experience: a phenomenological critique*, Cambridge University Press, 1983.

[Rol77] R. Roll, A Critique of the Asset Pricing Theory's Tests: Part I. *Journal of Financial Economics*, 1977, 4, pp.129-176.

[Rol82] L.T.C. Rolt, *Red of Danger*, Pan Books, Fourth edition, 1982.

[Rol84] R. Roll, A Simplified Implicit Measure of the Effective Bid-Ask Spread in an Efficient Market, *Journal of Finance* Vol. 39 (1984), pp.1127-1139.

[Ros76] S. Ross, The Arbitrage Theory of Capital Asset Pricing, *Journal of Economic Theory*, 1976, 13, pp.341-360.

[ROS96] L. J. Rosenblum, Projects in VR: Industrial Applications of VEs, *IEEE Computer Graphics and Applications*, 1996.

[RR00] S. Rasmequan and S. Russ , Cognitive Artefacts for Decision Support, In *Proc. IEEE SMC 2000 Cybernatics Evolving to Systems, Humans, Organizations and their complex interactions*, Tennessee, USA, October 8-

11, 2000.

[Rya97] T. W. Ryan, *Distributed Object Technology concepts and applications*, Hewelett-Packard Company, Prentice Hall PTR, NJ, 1997.

[SAA99] A. P. Sheth, W. Aalst, I. B. Arpinar, Processes Driving the Networked Economy, *IEEE Concurrency Journal*, July-September 1999, pp.18-31.

[Sal87] G. Salaway, An organizational learning approach to information systems, *MIS Quarterly*, 11(2):245-264, 1987.

[Sau97] V.L. Sauter, *Decision Support Systems: An Applied Managerial Approach* New York: John Wiley & Sons, 1997.

[SB98] P.H. Sun, W.M. Beynon, Empirical Modelling: A New Approach to Understanding Requirements, In *Proc. ICSEA Conference*, Vol. 3, 1998, Paris.

[SBK00] D. H. Sonnenwald, R. Bergquist, K.L. Maglaughlin, E. Kupstas-Soo, M.C. Whitton, Desiging to Support Scientific Research Across Distances: The nanoManipulator Environment, *Collaborative Virtual Environments*, London, Springer Verlag, 2000.

[Sed83] R. Sedgewick, *Algorithms*, Addison-Wesley Publishing Company, 1983.

[Sed90] R. Sedgewick, *Algorithms in C*, Addison-Wesley Publishing Company, 1990.

[Sha00] A. Shah, *Share Trading Game*, Third year CS project, department of Computer Science, University of Warwick, 2000.

[Sha64] W. Sharpe, Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk, *Journal of Finance*, 1964, 19, pp.425-442.

[Sha92] S. C. Shapiro, *Encyclopedia of Artificial Intelligence*, Wiley, New York, 1992 eds.

[She98] C.R.Sheth, *An investigation into the application of the distributed definitive programming paradigm in a teaching environment: The development of a virtual electrical laboratory*, MSc. Department of Computer Science, University of Warwick, UK, 1999.

[Sho90] Y. Shoham, Agent-Oriented Programming, Report # STAN CS 90 1335, Dept of Computer Science, Stanford University, October 1990.

[Sim60] H.A. Simon, *The New Science of Management Decision*, Harper & Row, New York, 1960.

[SKo98] M. Shader, A. Korthaus, Modelling Business Processes as Part of the BOOSTER approach to Object-Oriented System Development Based on UML, In *Proc. The Second International Enterprise Distributed Object Computing (EDOC'98) Workshop*, 3-5 November 1998, La Jolla, California, USA, PP.56-67, 1998.

[Slo90] A. Sloman, Must intelligent systems be scruffy?, In *J.E. Tiles, G.T.McKee, and G.C. Dean (eds), Evolving Knowledge in natural science and artificial intelligence*, Pitman Publishing, 1990.

[SMe88] S. Shlaer, S. Mellor, *Object Oriented Systems Analysis: Modelling the world in data*, Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ. , 1988.

[SMe92] S. Shlaer, S. Mellor, *Object lifecycles: Modelling the world in states*, Yourdon Press, Prentice Hall, 1992.

[Smi97] A. Smith, *Human computer factor: a study of users and information systems*, McGraw-Hill, 1997.

[Som95] I. Sommerville, *Software engineering*, 5th edition, Addison-Wesley, 1995.

[Son93] D.H. Sonnenwald, *Communication in Design*, PhD thesis, The State University of New Jersey, 1993.

[Son99] D. H. Sonnenwald, Evolving Perspectives of Human Information Behavior: Contexts, Situations, Social Networks and Information Horizons, In *Proc. Information Seeking in Context'98*, 1999, London.

[SPo00] P. Stevens, R. Pooley, *Using UML Software Engineering With Objects and Components*, Addison-Wesley, 2000.

[SS95] J. Salasin, H. Shrobe, Evolutionary design of complex software (EDCS), *Software Engineering Notes*, 20(5):18-22, 1995.

[SSD98] W. Skok, T. Sims, H. Doeringer, Design and Implementation of Enterprise Resource Planning (ERP) Systems using component-based Standard-Software, In *Proc. BIT 99 Conference*, Manchester, UK, 1998.

[STM95] P. Sallis, G. Tate, and S. MacDonell, *Software engineering: practice,*

management, and improvement, Addison-Wesley, 1995.

[STR98] P.A. Strassmann, What is Alignment?, *Cutter IT Journal*, Vol. 11, No.8, 1998, pp.4-9.

[Suc87] L.A.Suchman, *Plans and situated actions: the problem of human-machine communication*, Cambridge University Press, 1987.

[Sun98] P. H. Sun, W.M. Beynon, Empirical Modelling: A New Approach to Understanding Requirements, *In Proc. of ICSEA Conference*, Vol. 3, 1998, Paris.

[Sun99] P. H. Sun, *Distributed Empirical Modelling And Its Application to Software System Development*, Phd thesis, Department of Computer Science, University of Warwick, 1999.

[Sut89] J.W. Sutherland, *Towards a Strategic Management and Decision Technology*, Kluwer, Dordrecht, 1989.

[SWH99] K. S. Shrivastava, M. S. Wheater, Workflow Management Systems, *IEEE Concurrency Journal*, July-September 1999, pp.16-17.

[SYa98] S.A. Shumsky, A.V. Yarovoy, Self-Organizing Atlas of Russian Banks, In *Visual Exploration in Finance with Self-Organizing Maps*, Springer 1998.

[TA83] L.R. Talluru, V. Akgiray, Problem representation in decision support systems: an illustration from financial investments, In *Proc. Annual Meeting: Decision Sciences Institute*, San Diego, 1990, pp.481-83.

[Tay98] D.A. Taylor, *Object Technology.* Addison-Wesley. Harlow, 1998.

[TDF98] N. Taylor, D.V. Dijk, P.H. Franses, A. Lucas, SETS, arbitrage activity, and stock price dynamics, Warwick Business School, 1998.

[Tho97] R. L. Thomas, *Modern Econometrics: An Introduction*, Addison-Wesley Publishing, 1997.

[TL96] R.R. Trippi, J. K. Lee, *Artificial Intelligence in Finance and Investing*, IRWIN publishing, 1996.

[Tre65] J. Treynor, How to rate management of investment funds. *Harvard Business Review*, 1965, 43, pp.63-75.

[Tre73] J. Treynor, F. Black, How to use security analysis to improve portfolio

selection, *J.Bus.*, 1973, 46, pp.66-86.

[Tru96] S.V. Truong, *Interfacing Eden with Oracle*, Third year computer science project, Department of Computer Science, University of Warwick 1996.

[TW99] C. Taylor, D. Walter, E-commerce directions, *Internet Business Magazine*, May 99, Issue 28, pp 36.

[Twe99] J. Twentyman, Market Leaders, *Computer Business Review*, Special Report Series, ERP The Next Generation, 1999, pp 9-13.

[Vai96] R. Vaitilingam, *The Financial Times guide to using the financial pages*, 3rd ed., London : FT/Pitman, 1996.

[VEa98] J. Vince, R. Earnshaw, *Virtual Worlds on the Internet*, IEEE Computer Society Press, 1998.

[VF87] J.R. Valusek and D.G. Fryback, Information requirement determination: obstacles within, among and between participants. In *R. Galliers (ed.), Information analysis: selected readings*, pp. 139-151, Addison Wesley, 1987.

[Vin00] H.D. Vinod, Review of Gauss for Windows, including its numerical accuracy, *Journal of Applied Econometrics*, 15:211-220 (2000), John Wiley & Sons Ltd.

[VRWH94] R. Vidgen, J. Rose, B. Wood, T. Harper, Business Process Re-engineering: The need for a methodology to revision the organization, In *Glasson et al*, 1994), pp.603-612.

[Wal00] M. Waldron, Records Management: Has Been or Has to Be?", *Conspectus, E-process, Workflow & Document Management Technology*, September 2000.

[Wal88] J. Walmsley, *The New Financial Instruments, An investor's Guide*, John Wiley & Sons, 1988.

[WC88] L.B. Wilson, R.G. Clark, *Comparative Programming Languages*, Addison Wesley, 1988.

[Web7_1] http://developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm

[Web99-00] N. Webber, Financial Instruments: MSc lecture notes and computer

workshops, Warwick Business School, Summer 99, Spring 2000.

[Win84] P.H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, MA, 1984.

[Win93] R. Winder, *Developing C++ Software*, Second Edition, 1993.

[Woo00] J. Woodforth, *Interactive VRML front-end to dtkeden*, 3rd year project, Department of Computer Science, University of Warwick, UK, 2000.

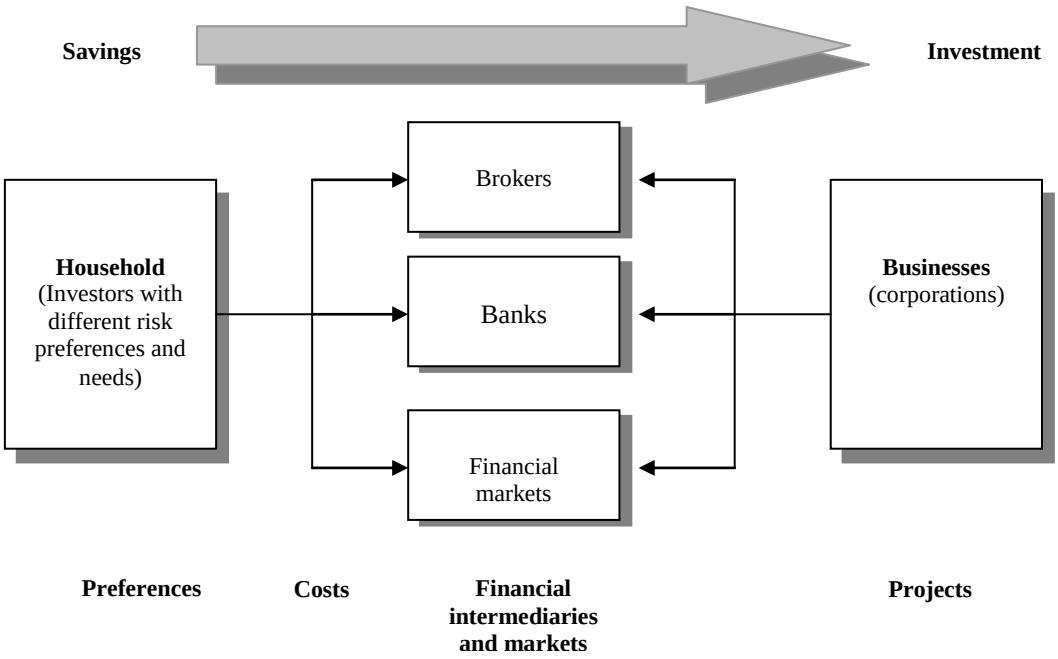
[Woo92] B. Woolley, *Virtual Worlds*, Penguin Books 1993, Blackwell 1992.

[Wyl91] M.F. Wyle, A Wide Area Network Information Filter, In *Proc. The First International Conference on Artificial Intelligence Application on Wall Street*, IEEE Computer Society Press, October 9-11, 1991, New York.

[Xia98] F. Xia, What's wrong with software engineering research methodology, *ACM Software Engineering Notes*, 23(1):62-65, 1998.

[Yun92] Y. P. Yung, *Definitive Programming: A paradigm for Exploratory Programming*, PhD thesis. Department of Computer Science, University of Warwick, 1992.

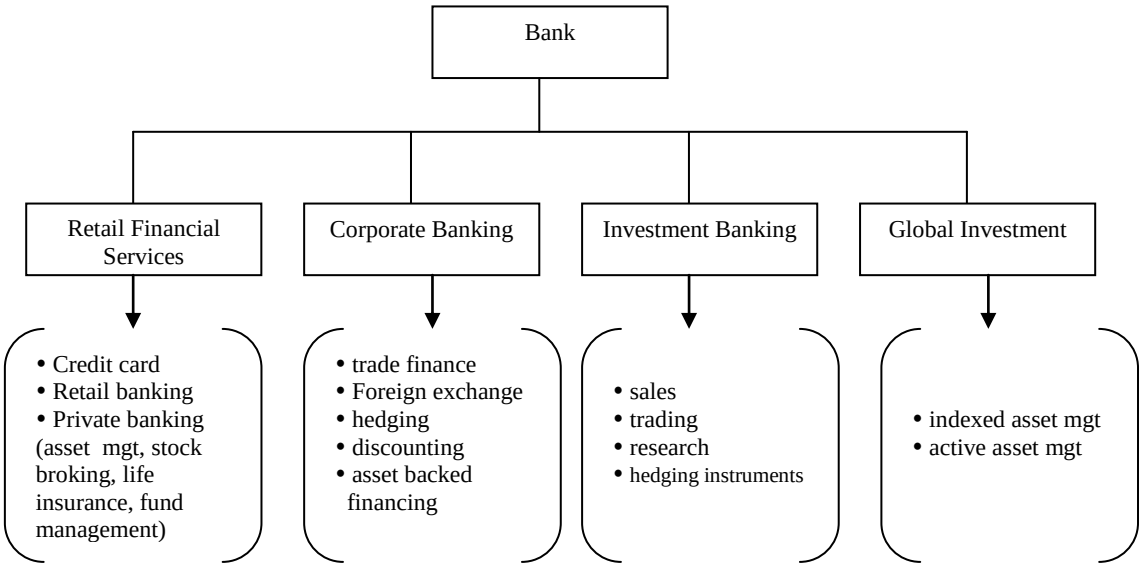
FINANCIAL INTERMEDIATION



Savings are transformed into investment in an economy via financial intermediaries (such as brokers and banks) and stock exchanges. The main role of financial inter-mediation is the reduction of the cost and facilitation of the matching process of borrowers and lenders needs, in term of amounts and time, and risk preferences.

Financial intermediaries conduct a special type of analysis to run their business and satisfy their customer needs. This analysis encompasses market prediction, portfolio management, risk management, and financial product development.

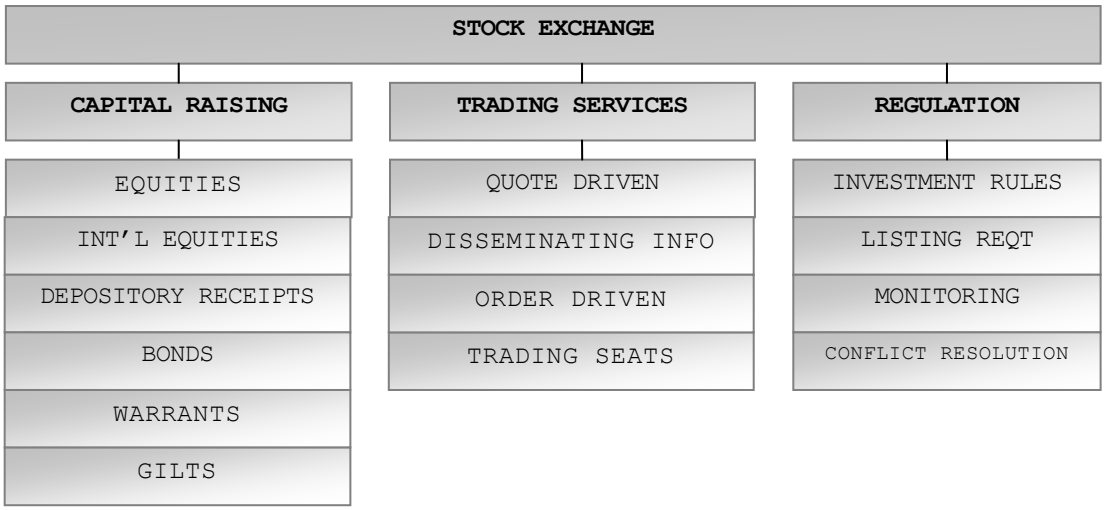
ORGANIZATIONAL STRUCTURE OF A BANK



ORGANIZATIONAL STRUCTURE OF A STOCK EXCHANGE

A stock exchange fulfils three basic functions:

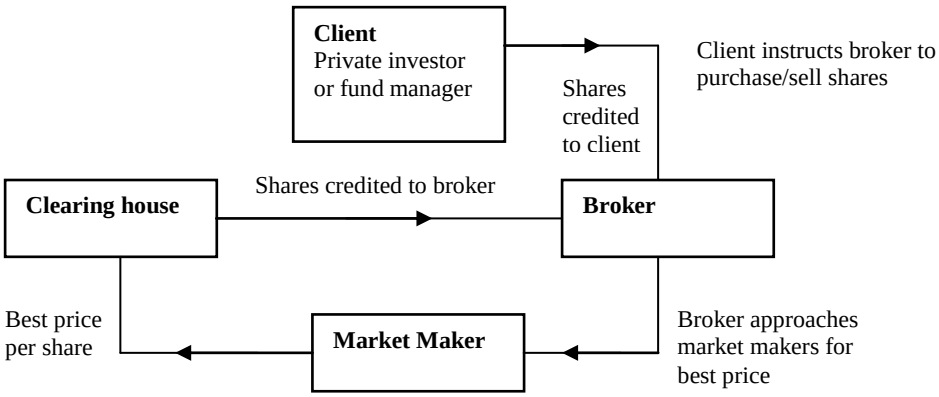
1) capital raising: allowing small and large companies to raise capital and to have their shares more publicly traded,
2) offering trading services: operating different types of trading systems – such as order driven, quote driven, or trading seats - and disseminating market information,
3) and regulating both capital raising and trading markets: through investment rules, companies listing requirements, exchange rules, and by monitoring members and companies compliance with rules, and resolving conflict of interest and rules violations.



The basic functions of a stock exchange [LSE]

A CLEARING SYSTEM

A clearing system helps financial intermediaries in the clearing and settlement of trades. The following figure depicts a typical flow of a transaction issued by a client and executed by a financial intermediary (broker) after clearing and settlement.



A clearing system [EASDAQ]

TOOLS FOR THE FINANCIAL ENTERPRISE

The integrated banking suite

Banks are increasingly offering online trading and banking services. Customers can login to the trading and banking services home page subject to security constraints and execute orders, view their accounts, monitor their portfolio, fill applications, and post inquiries. Electronic data interchange standards such as SWIFT, FIX, and OFX support online banking and trading.

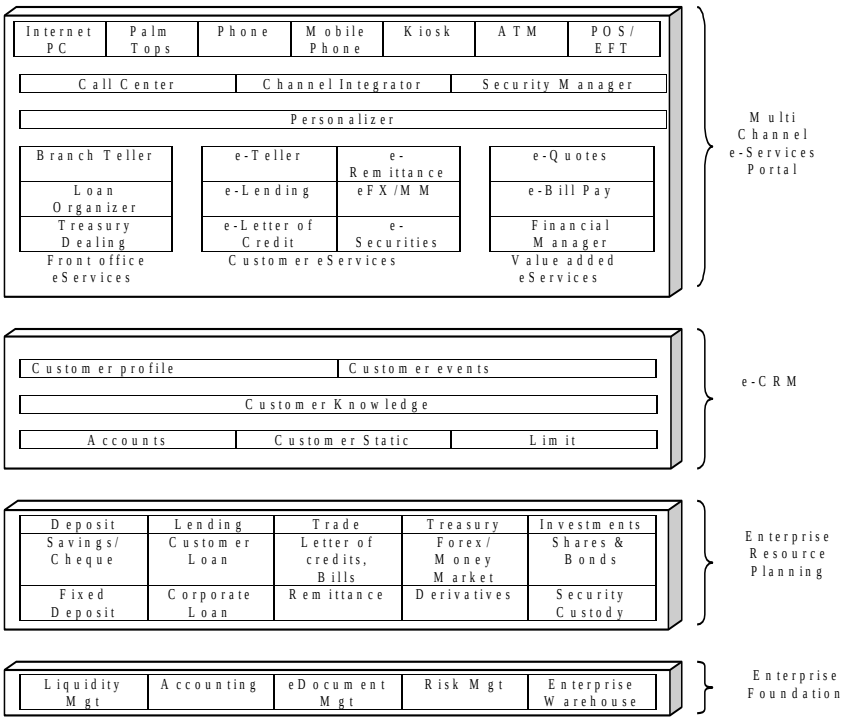


Figure 2.9 e-Banking operations

The general features of computer tools for automating banking operations are well exemplified by a suite of tools offering electronic banking services such as depicted in Figure 2.9. The Enterprise Resource Planning (ERP), multi channel integration, customer relationship management (CRM), electronic document management (EDM), and enterprise foundation tools for data warehousing that constitute such a suite are overviewed below.

• *Enterprise Resource Planning tools:* Enterprise Resource Planning is an umbrella term for all tools and technologies used to handle the internal operations of a banking firm (deposit, lending, trade, treasury, investment) and to automate its processes. The term back office refers to the IT centre where all Enterprise Resource Planning applications are handled. ERP software used to run on mainframe, however, the advent of the year 2000 problem, and the introduction of the euro currency is forcing all business and financial institutions to upgrade the tools and technologies used in their back-offices and to re-engineer their internal processes. The shift towards a client server architecture is accelerating, and the role of Enterprise Resource Planning software tools is not limited any more to handling the internal operation in a centralized location in the firm but is extended to manage the operations over

the firms multinational network and to integrate with its online and front-end applications. Major service providers of ERP tools include SAP[1].

Customer Relationship Management

The deregulation and the reduction of barriers to entry in the financial industry has resulted in the establishment of new institutions offering retail banking services at competitive prices and quality. Realising the need for new means to remain competitive in a global market place, top manager in banks and financial institutions started to shift their focus of interest from cutting operational cost to improving customer services and customer relationship management. Customer relationship management is referred to as the ability to capture a customer and to satisfy all their needs and requirements with minimum cost and high efficiency. This relies on many tools and technologies which can capture all the relevant information about a customer and their need via multiple channels (e.g. telephone, fax, internet), store this information in databases, and analyse it using data-mining and business intelligence tools. The CRM concept has matured and grown to become a subject encompassing all aspects of interaction between a company and its customers. It encompasses client servicing, targeting, profitability, selling, distributing channels, e-commerce, sales strategy, product strategy, and much more.

Electronic Document Maanagement

This range of tools helps in storing, retrieving and managing the workflow of documents across the enterprise, and in undertaking collaborative work. It is very important for banks and financial institutions which have large amounts of important paper documents such as faxes, credit application, checks, financial statement, etc.. Documents are not stored in one place and are not processed by only one person. They have a workflow, that is a document is passed from one person to another across the enterprise for approval, amendment, or rejection. Automating this workflow is an important task to consolidate financial reporting and reduce the accounting errors. Collaborative work on projects such as syndicating financing, co-financing, and financial research requires the storage of documents and the management of their workflow. This task is facilitated using specialized document management tools for project collaboration.

[1] http://www.sap.com/

Multi-channel integration tools

Survey data has shown that once e-commerce efforts were undertaken, consistency across multiple delivery channels became the greatest implementation challenge[2]. Key issues addressed in multiple channel integration are: choosing the appropriate channels, providing customers with good service across all channels, ensuring consistent information access across all channels, organizing a centralized policy to co-ordinate everything, and building and maintaining customer confidence. Tools used to ensure successful multi channel integration include tools for the data warehouse, computer-aided business process re-engineering tools, and network management tools.

Business Process Modelling

Business process modelling tools support the re-engineering of the financial enterprise. Models used to document business processes and operations are often subject to standards (the US Federal Information Processing Standard IDEF0, the ISO9000 documentation for the manufacturing industry, etc.). Today organisations are facing a continuous change, and the effect of this change on business efficiency is highly unpredictable and risky. Simulation tools has been developed to take the risk out of change in several ways: by evaluating the impact of adding a new product line, by re-engineering the business processes, by modelling the working environment and by anticipating the implications of different business decisions.

Data-warehousing tools

A data warehouse contains a collection of data from various operational systems and sources. It can be used as an integrated information base for making decisions and solving problems. Data-warehousing is the process of integrating enterprise wide corporate data into a single repository supporting a variety of decision analysis functions as well as strategic operational functions. The data warehouse provides a link between the operational environment and analysis environment of an organization. Tools for building, maintaining, and using the data warehouses in the financial enterprise are provided by Digital (Compaq)[3] and Informix[4].

---

[2] Ernst & Young survey cited in Technology in Banking and Finance.
[3] http://www.digital.com/alphaserver/bus-intell/index.html
[4] http://www.informix.com

TOOLS FOR INVESTMENT

FINANCIAL ANALYSIS AND MODELLING

💻 OLAP

General purpose OLAP (Online Analytical Processing) tools are a common example of business intelligence tools. Data rich industries (such as the financial, marketing, consumer goods production, services, and transport) have large quantities of good internal and external data available in their databases to which they need to add value in order to gain competitive advantages in a global marketplace. OLAP tools have been developed in response to this need, to help conduct sales and customer behaviour analysis, budgeting, financial reporting and consolidation, tailored and intelligent management reporting, and performance measurement. However, the success of these tools relies on a high analytical functionality, large data capacity, and an underlying integrated database. The efficiency of OLAP tools might be reduced drastically in firms operating with multiple databases and low quality data (lacking consistency, reliability, accuracy, and integration).

💻 Data Mining

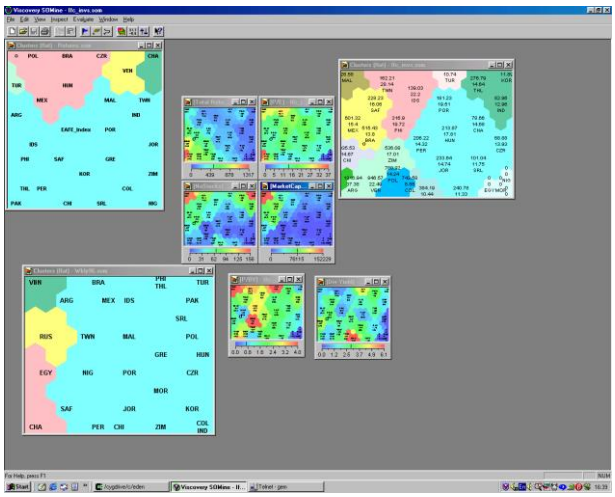
Many databases contain valuable information that is not readily obvious. An example of unrevealed information might be patterns of high-risk companies within a financial database. The search for these valuable, yet hidden, patterns and relationships within a database is known as data mining. Data mining infers rules that can guide decision making and forecasts the results of the decision. A number of different types of data mining tools are in use today, including: data visualization, neural networks, decision trees, and rule induction programs. These tools help in visualizing, detecting patterns of relationship, and inferring business rules hidden in the underlying database. The source of data analyzed by data mining tools can be loaded from ODBC compliant databases as well as spreadsheet and statistical software packages. Some of these tools can be also integrated into applications as activeX components. The features in data mining tools include: data manipulation (sampling, selecting, and merging data sets); modelling (classification, prediction, profiling, clustering, and detection models); visual exploration (visual presentation of the different stages of the data mining process, charts and graph plotting, and graphical trees views), and online distributed reporting.

Spreadsheets

Spreadsheet modelling is widely used in business and finance. The ease of use, functionality, and built-in features make spreadsheets the most popular and basic tools for simple financial analysis and graphics reporting. Spreadsheet data models are the underlying models used in any intelligent tool with extended analytical capability. Almost all financial analysis and visual exploration tools import their data sources from spreadsheet applications.

Visual Exploration

Visual exploration tools for building Self Organized Maps are used in finance to perform credit scoring, risk assessment, behaviour modeling, knowledge discovery in data bases, system state monitoring, process engineering, quality control, and prediction. This range of tools supports: dependency analysis, deviation detection, unsupervised clustering, non-linear regression, data association, pattern recognition, animated monitoring, as well as other enhanced visualization techniques.



Visual exploration in finance using self organized maps

Visual exploration, as its name indicates, explores the data set without needing to make any prior assumption about its characteristics. As such, it offers an open-ended potential for exploring the data set. Testing the significance of the results of a visual exploration method is not formalized, and we are as yet unable to judge the relevance of our analysis. Visual exploration has to prove itself as a reliable means for analysis.

This section considers the self organized map (SOM) as an example of a visual exploration technique and tool used in finance. SOMs have been used to understand trends and patterns among today's emerging market [Deb98], to analyse data on the emerging Russian banking system [SYa98], to translate multi-dimensional mutual fund data into simple two-dimensional maps [Deb98-2], to approximate the distribution of the interest rates structure and its

deformations over time [BGC98], and to conduct many other financial case studies where no econometric assumptions on data can be made.

The SOM technique has found a wide application in finance because it is a numeric rather than a symbolic method, it is a non parametric method, it demands no *a priori* assumption about the distribution of the data, and it can detect unexpected structures or patterns by learning without supervision. One of the tools used to develop SOM is Viscovery SOMine[5].

Risk Management

The risk of a financial asset is defined as the variation in its underlying value. Credit granting, investing and trading involve risk. Hedging is a way to protect against risk, it can be achieved by the use of derivative financial instruments which are mainly financial contracts that can guarantee a non-loss position if all possible adverse market conditions and price movements are taken into consideration in this contract. Risk management involves the detection of trends and market surveillance. Self organized maps are used in visualizing some risk assessment and classification problems, such as the classification of countries according to their risk, where the level of risk is evaluated in term of many economic and financial factors.

Market Prediction

Analyzing time series data in order to recognize patterns or make prediction about future values is important in many application area such as investment, trading, etc.. The ability to predict future values based upon past values and known future events is implemented using statistical time series analysis or neural networks[Bow90].

Portfolio Management

A portfolio is the total securities held by an institution or a private individual. Voluminous amounts of rapidly changing data in financial markets create a challenging problem for portfolio managers attempting to exploit such changes to achieve their investment objectives. Changing market conditions should be exploited to optimize the value of individual portfolios [Pat90]. Prices of financial assets taken from reliable sources are fed into portfolio management models which support the decision of the buy and sell action and adjust the portfolio holding according to current prices. Portfolio management tools enable the user to download list of prices for stocks, options, bonds, mutual funds, or other investments directly

[5] Free demo available at http://www.eudaptics.co.at

from the Internet. Based upon the newly downloaded prices a portfolio is priced and reports are generated to display the results in ways that enable the investor to make clear and precise investment decisions based on the total portfolio behaviour. Appropriate visualization is used to highlight profitable transaction based on a study of asset prices, and to show the portfolio holding at any moment in time.

Numerical Libraries

The implementation of intelligent financial analysis is supported by libraries developed by commercial firms. These libraries include a wide range of financial computation functions. Spreadsheet applications also support a library of financial and statistical functions. However, human judgement is always necessary to conclude a final judgement on the investment decision.

Statistical Packages / Fourth Generation Languages

Statistical tools extend the features of spreadsheet tools to encompass a wider range of numerical and econometric techniques implemented as built-in functions. Statistical packages are used widely in academic financial research.

Fourth generation languages (4GLs) [WC88] were developed in response to the dissatisfaction of business users with large conventional languages like COBOL. These users are not professional programmers and wish to obtain quick results from data stored in a computer. There is a vast variety of 4GLs including spreadsheets, application generators, query languages, and decision support languages. Fourth generation tools for statistical and econometric analysis include: Excel, Gauss, Matlab[6], Eview, minitab, and SPSS[7], Mathematica[8], Stata[9] statistical software, etc..

An important issue considered in the use of fourth generation tools for statistical and econometric analysis is numerical accuracy [Vin00]. Fourth generation tools for statistical and econometric analysis are usually evaluated for their flexibility, power, intuitive syntax, ease-of-sharing, and ease-of-use. Fourth generation tools have a variety of different features, consist of different components, have different syntax and notations, and require different learning time. For example, the GAUSS system consists of three components: the GAUSS programming language, a publication quality graphics library, and a library of applications

[6] http://www.mathworks.com/
[7] http://www.spss.com/
[8] http://www.wri.com/

modules. Learning GAUSS takes about 2 weeks, while learning Excel might take hours for a computer literate person.

TOOLS FOR INVESTMENT

EDUCATIONAL

💻 OPTION! SOFTWARE

OPTION! is an educational package accompanying a book on options [Kol96]. The package is command line driven and consists of nine modules. Its procedures can compute virtually all of the model prices and examples given in the text book [Kol96]. Where applicable, the program can also graph option relationships. The first module, Option Values and Profits at Expiration, computes the outcome for various option strategies that are held until expiration of the option. Reports can be viewed or printed summarizing the results of the considered strategy. The second module, Option Values and Profit Before Expiration, is similar to the first module, but allows the user to price the options in their portfolio by using Black Scholes and Merton Models. Results can be viewed or printed and graphs can be plotted. The Third Module, European Stock Option, is composed of six sub-modules, covering the binomial model with specified price movements, the Black Scholes model, implied volatility according to the Black Scholes and Merton Models, simulation of stock and option prices consistent with the Black Scholes model, the binomial approximation of the Black Scholes model using stock prices movements, and dividend adjustment for European options. The fourth module, American Stock Options, consists of five sub-modules that cover virtually all dimensions of pricing American options on individual stocks. The fifth, sixth, and seventh, module cover Stock Index Options, Options on Futures, and Foreign currency Options. The eighth module deals with exotic options. The last module, Cumulative Unit Normal Probabilities, computes univariate and bivariate cumulative probabilities.

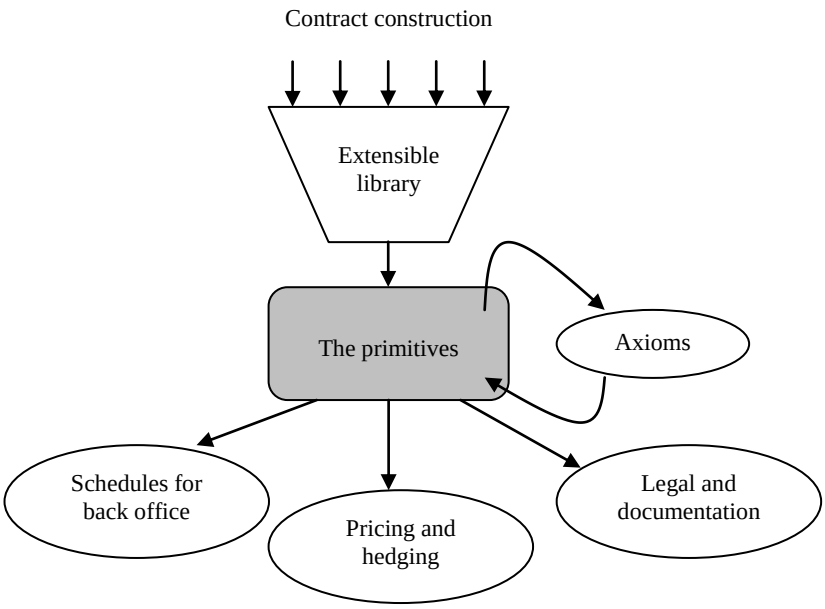
[9] http://www.stata.com/

TOOLS FOR INVESTMENT

FINANCIAL ENGINEERING

💻 Special Purpose Tool for Financial Contracts Description

The following sections describe joint research work at Société Général in Paris, and Microsoft Research in Cambridge, led by Jones et al (2000). This work attempts to develop a computer language for the description and valuation of financial contracts. The following paragraphs overview the proposal of Jones et al for a new "language of contracts" intended to precisely specify arbitrarily complex contracts, to build easily-extensible libraries of contracts, to perform valuations of complex contracts in a simple, modular way, and to perform other back office functions. The key idea behind this research is that complex contracts are formed by combining together simpler contracts, which in turn are formed from simpler contracts. In that context three keywords are introduced, the first is *combinators*, which refers to *bricks* and *mortars*. *Bricks* are elementary contracts from which all others are built (e.g. receive £100 on 1 Jan 2004). *Mortars* are the ways of transforming and combining contracts to make more complicated ones. According to Jones et al (2000), defining contracts using a fixed set of combinators, is similar to the specification and implementation of programming languages. A programming language has primitive elements (variable, constants, etc), and combining forms (for loops, if-then-else, procedures). The meaning, or behaviour of a complex program can be explained in terms of the meaning, or behaviour of its component pieces). The main research outcome is the definition of a carefully-chosen set of combinators which can be used to describe many familiar contracts as well as processing these contracts (by giving an abstract valuation semantic to combinators). A compositional approach is used for pricing and hedging purposes. For this purpose, a formalized language to describe financial contracts is being developed with the aim of facilitating back office contract execution, accelerating in a substantial way risk analysis, graphically representing a contract as a decision tree, reminding the front-office about any potential exercise decision that has to be taken, and generating intuitive simulations for the marketing departments of investment banks. The following picture, reproduced from Jones et al (2000), illustrates the main features of the formalized language to describe financial contracts.



Features of the formalized language to describe financial contracts proposed by Jones et al (2000)

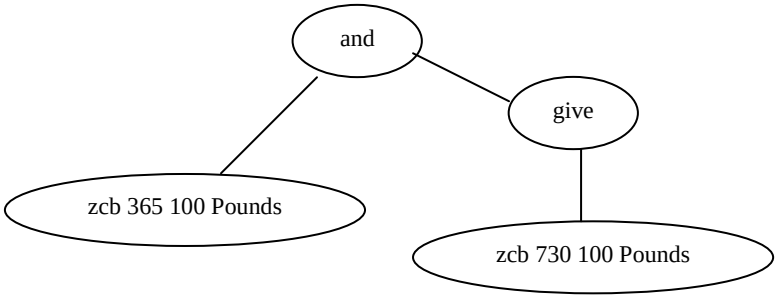
In describing financial contracts, a precise vocabulary is used. This includes the following terms: rights and obligations, observables (a time varying value whose value at any time is a matter of legally-enforceable fact e.g. LIBOR rate, the temperature in Dover), unit of currency (unit of currency in which a traded asset is measured), acquisition date (date of acquiring a contract), horizon (expiry date of the contract).

A notation is used by Jones et al (2000) to define financial contract, this notation is implemented in the functional programming language Haskell [JHA98]. A simple zero coupon bond contract is described using this notation as follows:

C1::Contract ←	Type signature for C1, that is C1 has type contract
C1=zcb t1 100 GBP ←	The application of the function zcb to three arguments
zcb :: Date → Float → Currency → Contract ←	Type signature for zcb

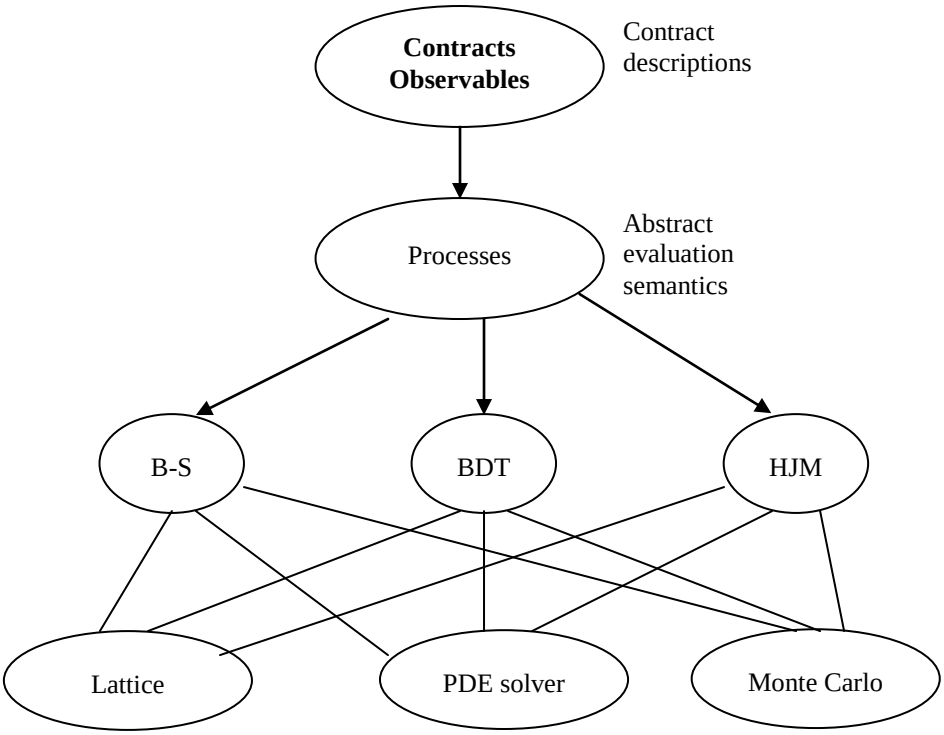
The description of the contract, c1, Zero coupon bond contract "receive £100 in 1st January 2010" using the Haskell notation.

Examples of combinators are **and**, **give**, **andGive**, etc…. The use of combinators (*bricks* and *mortars*) is illustrated in the tree description of the following contract composed of two *bricks* (elementary contracts) and two *mortars* (combinatory factors).



A contract as a tree [JES99]

For evaluating financial contracts two layers are used: abstract evaluation semantics (translate a contract into a value process) and concrete implementation (concrete implementation of processes). The following picture depicts the contract valuation framework proposed by Jones et al (2000).



Framework for contract valuation proposed by Jones et al (2000)

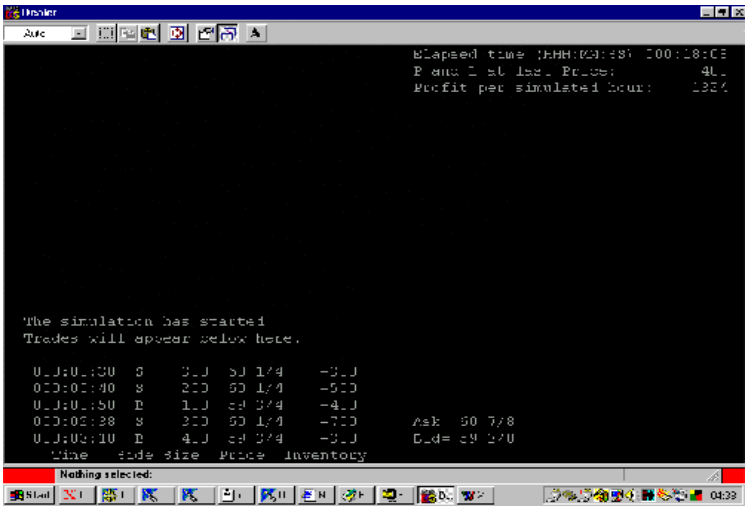
TOOLS FOR THE FINANCIAL MARKET

EDUCATIONAL MODELS

The Wall Street Trader

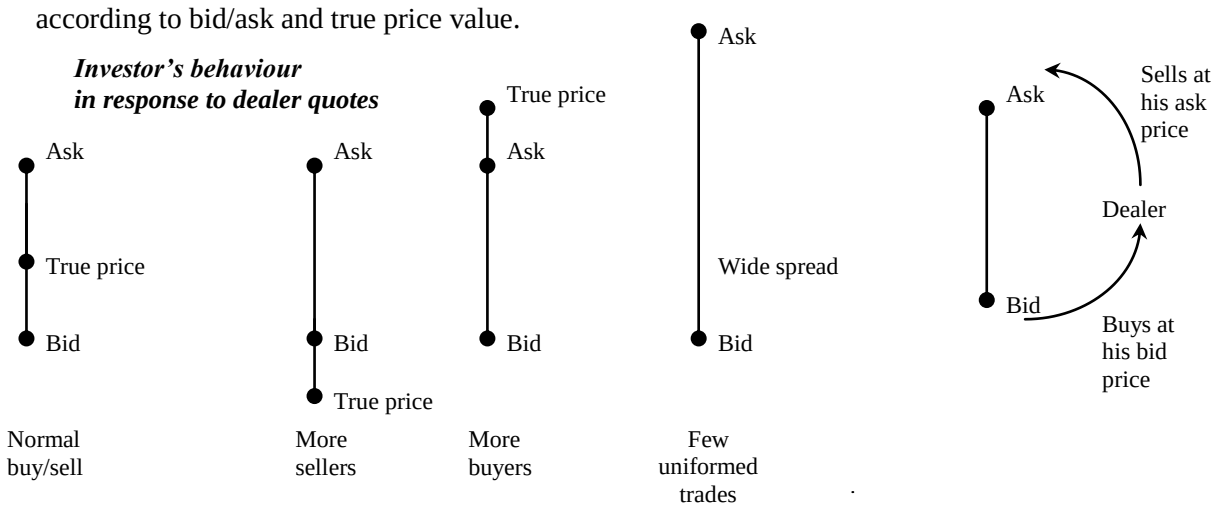
The Wall Street trader is a wimp based application game with multimedia support, to help learning about the global financial community. The game has a simulated financial database covering two years of trading history. The user, the player of the game, is the investor and he/she can buy and sell stocks and read about a company's history. Some tools are available to the investor: an analyst (to help the user understand the news), an insider (to help the user uncover hard to find facts), and a spy (to see what other investors are up to). The game has a tutorial. The news database is fictional though inspired by real events. Each time the game is restarted the events are changing so that the game can be played many times without being repetitive.

The Monopoly Dealer Simulation



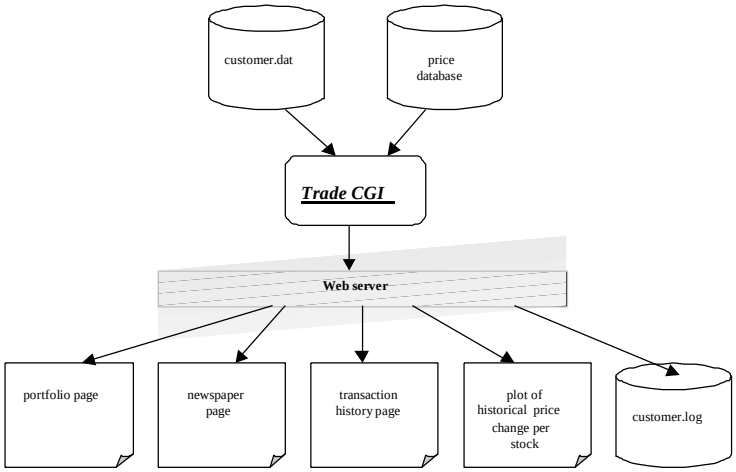
The monopoly dealer simulation is a command line application with a simple interface but a rich content. The simulation reveals the complexity of the interaction in the financial market and the difficulty in apprehending financial markets phenomena, such as the determination of the true price of a security and the impact of the investor behaviour (trader flow) on the decision making of a dealer.

The following description of the monopoly dealer[10] simulation, developed by Larry Harris, is extracted from the corresponding homepage. The program simulates trading in a dealer market in which there is only one dealer (the user of the simulation model). The user's task (the sole dealer) is to set and adjust bid and ask quotes (raise, lower quotes, or narrow and widen the spread) to maximize his trading profits. The computer model simulates traders arriving at random times to trade with the dealer (user) at his quoted prices. Larry Harris's aim in the simulation is to raise the awareness of its user (playing the role of a dealer) to the trading behaviour of different types of investors (informed/uninformed), and the true value of the security (changing through time and known to informed traders). The role of the user (dealer) is to estimate the true security value by examining the order flow. While the simulation is running, the computer estimates the user (dealer) profits by adding his cash position to the current market value of his inventory (computed using the last trade price). When quitting the simulation, the computer shows the true security value and the true profits of the dealer (user). The dealer (user) should know how to attract traders by adjusting his bid/ask quotes and spread. When the quoted bid/ask spread is wide, few uninformed traders will trade. To encourage uninformed traders to trade the spread should be narrowed. If the true security value is above the ask quotes, informed traders will buy from the dealer. They will sell to the dealer if the true security value is below his bid. Informed traders will trade more often and they will make larger trades when the dealer quotes are far from the true security value. The dealer should watch and control his inventory carefully, holding continuously less than 10,000 shares, long or short. The simulation will end if the dealer's inventory goes above 10,000 or below -10,000. Upon termination, the simulation reports executed trades. The following diagram depicts the buy/sell reaction of the informed investor according to bid/ask and true price value.



[10] http://lharris.usc.edu/trading/DealerGame\Default.htm

💻 Web-Based Trading Model



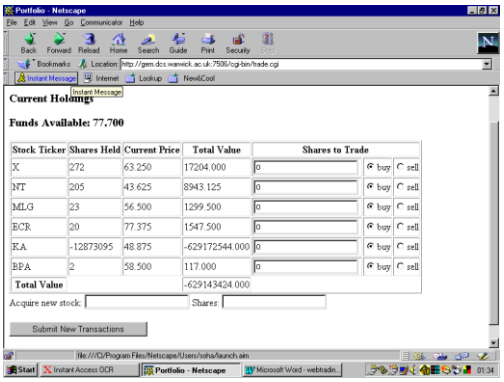
Boutell (1996) developed a web trading model using CGI (Common Gateway Interface) implementing a stock market trading system. As in online trading systems, the main features of this web-based model are to allow users (investors) to perform three tasks: 1) examine their portfolio; 2) buy and sell stocks; and 3) track the performance of stocks over time. The model does not work in real time (i.e. it has no feed of real data), however, it could be interfaced with a real database of stock market data with fast price updates and a large number of securities.

Boutell's model takes into consideration security, which is an important issue in designing any web trading or e-commerce application. Providing security means providing an appropriate authentication system. The model adopts a simple authentication mechanism to password-protect directories. It requires each user to enter a valid account name and password to access any page in that directory.

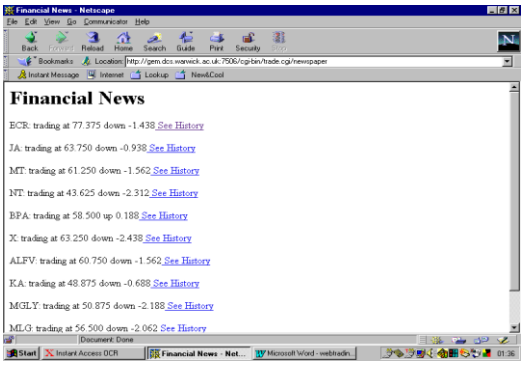
The web-trading model consists of two CGI programs. The first one generates random price fluctuations, simulating a one-day activity in the stock market. For real applications this CGI should be replaced with an interface to a source of actual stock prices. The second CGI program interacts with the user (which is the customer). It accepts user input and allows the customer to monitor his portfolio, to buy and sell shares, and to track the performance of shares. The CGI application developed to simulate stock price fluctuation accepts as input a text-based file consisting of stock ticker symbols. After running the application the database file will hold each stock ticker followed by up to 30 days of fluctuating stock prices, with the current price being the last one. The main functions of the CGI trade application is the dynamic generation of web page content from customer.dat files (each authenticated customer has a customer.dat file, e.g. the customer soha has soha.dat file), and the stock price file generated by the first CGI program. The dynamically generated web pages are: a portfolio page (listing the shares holding of a customer and giving this customer the option to buy and sell from already owned shares or to acquire new shares), a newspaper page (giving the

closing price of a share, i.e. the last generated price by the stock price generator CGI application, and the option to view a graphical plot of the price movement of each share), and a transaction history page (giving the time, date, and the number and symbol of shares bought or sold by a given customer).

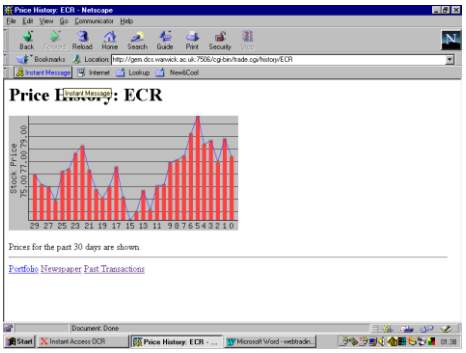
The web pages generated by the trade CGI program, are depicted in the following screenshots taken from the web trading application developed by Boutell (1996) and accessible at http:gem.dcs.warwick.ac.uk:7506/cgi-bin/trade.cgi.



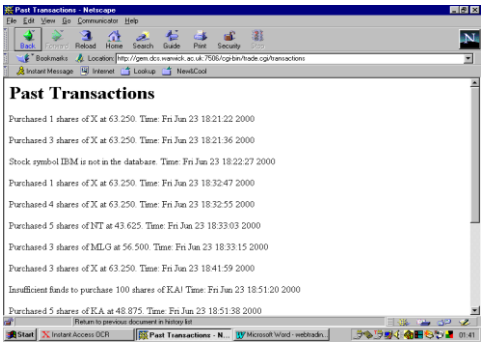
Generated portfolio page



Generated newspaper page

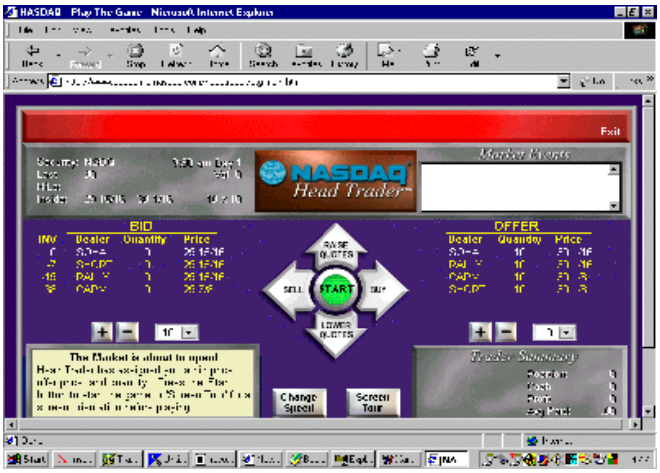


Generated graphical plot of historical price changes for a given stock



Generated transactions history page

Head Trader



Head Trader[11] is a web-based educational simulation developed by the Nasdaq Stock Market®. It is based on an original model and software developed by two academics in the field of trading mechanisms - Robert A. Schwartz and Bruce W. Weber[12]. Head Trader simulates the experience of a Nasdaq Market Maker buying and selling stocks in a screen-based market environment. It puts the player in the shoes of a professional trader. The game interface and information streams have been designed to be similar but not identical to the Nasdaq Workstation II used by Nasdaq's more than 525 market-making firms.

In a Solitaire Play, players can compete against computerised players for bragging rights as the best sell-side trader. The player is given a chance to react to events that occur in the market such as changes in supply and demand, news, and the actions of computer Market Makers. These reactions determine the player's success.

In Competition Play, players participate in various competitions and, in some instances, win prizes. It also allows professors and other instructors to hold private competitions for their students. The best scores for each competition are posted to scoreboards until a better trader comes along.

Upon termination of the simulation, a performance measurement, or score is displayed. The game time varies depending on the skill level selected and the amount of user interaction.

STOCK TRAK

STOCK-TRAK[13] is an investment simulation, offering its users the opportunity to gain practical experience trading a wide range of investment vehicles. STOCK-TRAK users have

[11] http://www.academic.nasdaq.com/headtrader/
[12] at the Zicklin School of Business at Baruch College, The City University of New York
[13] www.stocktrak.com.

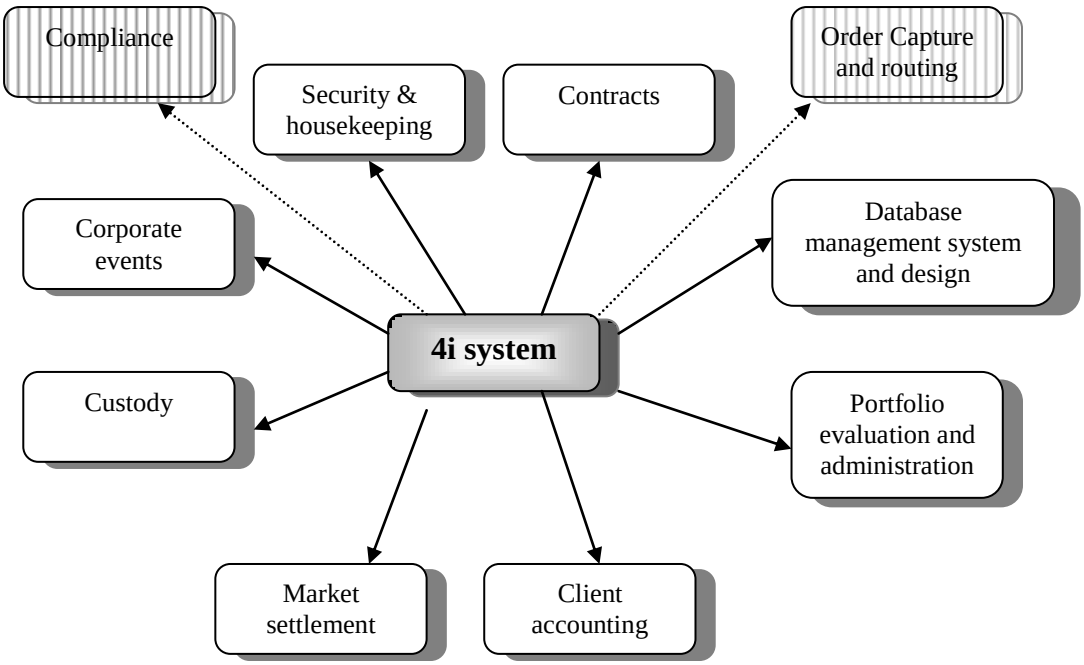
$100,000 to invest in common and preferred stocks, bonds, options, index options and futures, commodities, foreign currencies, financial futures, options on futures, spot contracts, mutual funds, as well as international stocks. Other features of the simulation include day trading, buying on margin, writing options covered and naked, as well as short selling. Interest is earned on cash balances and incurred on margin balances. STOCK-TRAK offers toll free access to live brokers, as well as twenty-four hour account access online. Users range from junior high school students to finance majors to firms that use STOCK-TRAK to train new employees.

TOOLS FOR THE FINANCIAL MARKET

TRADING SYSTEMS

💻 (4i) System

Trading, valuation, and management of financial instruments as used by practitioners is exemplified by the Integrated Investment Intermediary Information System (4i) developed by Consort securities systems Ltd.[14]. The system offers integrated investment management and stockbroking services. The system is WIMP based, with a GUI front-end and an SQL-Server relational database back end. The functionality of the system is detailed in the figure below.



The functionality of the 4i system

A single core database is used by all modules of the 4i system. This database maintains client / counterparty / principle and static data. Enquiries and reporting, batch and online processing, and multi currency are supported by this backend data base. Contracts are channelled to the system via the order and capturing module. The contract module allows the issuance of contracts (the layout of the issued contract can be user defined), the performance of calculations related to financial contracts, the reversal of contracts (cancellation or amendment) and settlement of the contract. The order and capturing module is designed to capture orders from a variety of sources (both manual and automatic), to validate them against specific criteria, and to direct them to the appropriate dealing system. The client accounting module is a look up module for investor, account, agent and brokers details. The results of the look up are reported through the financial reporting module. The portfolio evaluation and administration module allows the creation and maintenance of portfolios for clients. Client portfolio information comprises contracts, prices, indices, currencies, and valuations. In this module, portfolio modelling and what-if analysis can be conducted, stock modelling and asset allocation can be performed, capital gain tax can be calculated, and performance measurement and reporting facilities are also provided. The client accounting module manages the accounts of investors, agents and brokers and provides accounting reporting facilities. The market settlement modules report trades to settlement agencies and regulatory bodies, and monitor and record the progress of transaction settlement and trade execution. The custody module allows the creation and maintenance of new information records in the database. The compliance module provides reports for compliance officers to complete their tasks. The security and house-keeping module contains system configuration files storing licensing data, system program control info, and other features related to system maintenance, monitoring and control.

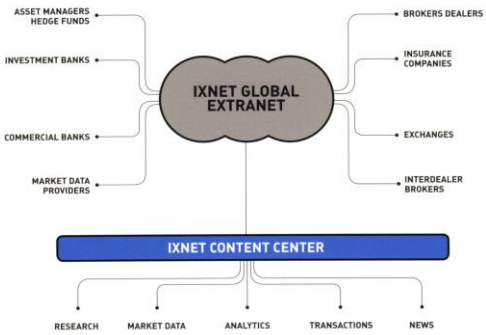
💻 eSpeed

Commercial software packages are developed to enable timely trade execution. An example of such type of product is the eSpeed[15] system that provides instantaneous, anonymous execution and trade confirmation. It offers a clear, comprehensive price display screen with real-time price dissemination.

[14] http://www.consort.co.uk
[15] www.espeed.com

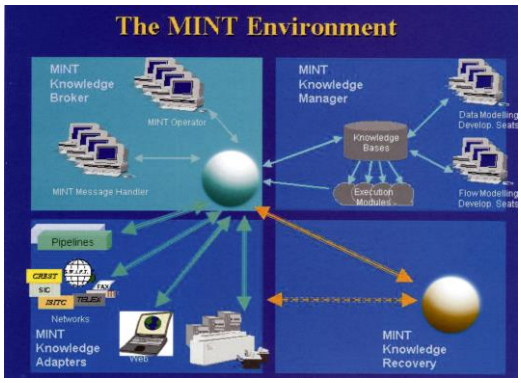
IXNET

IXNET global Extranet offers instant access to hundreds of financial services firms, fast and dependable electronic transaction services, critical market data, real-time industry news, in-depth analysis, and essential research.



Integrated Solution for Trading

Sungard[16] also provides integration modelling tools that allow message routing and flow to be defined using a graphical user interface (GUI).



Sungard integration solution

[16] www.sungard.com

SECURITY MARKETS

Security Markets are places where traders gather to trade securities (eg. LIFFE or Forex Market). *Trading* is a search process, where buyers or sellers try to find a counter-party. Price, quantity and time to the trade are key factors. *Dealers* or *brokers* help people trade. Dealers are willing to take the other side of a trade on demand. They quote a bid (buy) price and a offer (ask) price and profit from the spread. Dealers acquire their clients' positions and then try to trade for them at a profit. Brokers are agents who help traders search for counter parties; they profit through commissions. Security markets are designed to reduce counter-party search cost. Key elements which make markets work are: asymmetric information between informed and uninformed traders; order flow externalities "trade attracts trade!"; trading rules; communication and trading technology; arbitrage between assets in different markets; principal-agent issues[1]; trustworthiness and creditworthiness; and the legal and institutional framework.

CHARACTERISTICS OF FINANCIAL MARKET MICROSTRUCTURE

Three main characteristics define the financial market structure (cf. Figure A.3): a. trading sessions (time intervals at which trades take place), b. execution systems (matches buyers with sellers), and c. information systems (bring information into and out of the market.

a. *Trading sessions* differ across different types of markets. *Continuous markets* arrange trades continuously as orders arrive. *Call markets* collect orders for *batch* processing.

b. Markets are usually classified by their *execution systems* (the procedures for matching buyers and sellers), these include:

- *quote-driven systems*: are primarily organised by dealers (e.g. NASDAQ, London International Stock Exchange (SEAQ), OTC Bond markets, Forex markets). In a pure quote-driven markets, dealers supply all liquidity. Dealers quote their *bid* and *ask* prices. Better prices and larger quotes for larger sizes may be obtained through negotiation. Brokers or buy-side traders choose which dealer they trade with. Narrow spreads provide a measure of fairness;
- *order-driven systems (auction markets)*: are organised by exchanges and follows order precedence trading rules to match buyers and sellers and a set of *pricing rules* to

[1] Agents may not act in the best interests of the principal; brokers may not work as hard as you may want them to etc

determine the trade prices (e.g. Tokyo Stock Exchange; Paris Bourse; Toronto Stock Exchange). Since traders cannot choose with whom they trade, order-driven markets require *clearing houses*;

- *brokered systems*: are organised by brokers who actively search for matching buyers and seller. Brokered markets usually arise when the item traded is somehow unique and when dealers are unwilling to hold inventories. Brokered market examples include block trading market, market for ongoing concerns (businesses) and real estate market;
- *hybrid systems*: are a mixture of order driven, quote driven and brokered market. Hybrid systems are order-driven auction markets in which the specialist must provide liquidity under some circumstances. Many US stock and options exchanges have specialist systems.

c. Information Systems collect, organize, present, store, and transmit information about orders, quotes and trades. Electronic trading systems facilitate collection of information from market participant. *Order routing systems*, o*rder presentation systems, and order books* are used to transmit, present, and manage standing orders. Electronic order routing systems transmit standardised orders with great accuracy at low cost. These systems may be maintained by brokers, dealers or exchanges. Complex orders are often communicated by telephone. On some exchanges, hand signals may be used to send an order from an order clerk to a floor trader. An example of Order Routing System is NYSE's SuperDot. In *open outcry auctions (oral auctions)*, traders sell out their bids and offers on the floor of an exchange. In *screen-based trading systems*, orders are presented on computer screens. In *board-based trading systems*, orders are written on a big board. Order books hold orders that have not yet been executed. An order book may be an electronic database or a box of trading tickets. Brokers, exchanges and dealers may all maintain order books.

Collected information is distributed to member traders. *Market data systems* report trades and quotes to the public. *Price and sale feeds (ticker tapes)* report trade prices and sizes. *Quotation feeds* report quotation changes. The Securities Industry Automation Corporation (SIAC) maintains the Consolidated Trade and Quotation System (CTS and CQS). Trade information is sold to various data vendors who repackage it for distribution to the public.

A market is *transparent* when complete information is reported to the public quickly. A market is *liquid* when traders can trade when they want to without much impact on price.

Orders are instructions traders give to brokers and/or exchanges explaining how their trades should be arranged. Traders use orders to communicate their intentions. An exchange

arranges trades by applying rules for matching orders. The order submission strategy affects trading profits and liquidity. Different types of order can be issued, these affect market liquidity and execution price:

• *market orders:* instruct the brokers to trade at the best price currently available;

• *limit orders*: instruct the broker to trade at the best price available but to not violate the limit price;

• *stop orders*: activate only after price reaches some threshold called stop price;

• *market if touched orders:* are traded at the market price if it touches some preset price;

• *tick sensitive orders*: specify tick[2] conditions for trade execution.

Different type of instruction can be associated with an order:

- *Validity Instructions* indicate how long the order remains (*Good-till-cancel* orders remain open indefinitely; *Good-until* orders specify an expiration date; *Day orders* expire at day-end; *Immediate-or-cancel, good-on-sight* orders and *fill-or-kill* orders expire immediately following presentation)
- *Quantity instructions* indicate how large orders can be broken into small trades. *All-or-none* orders must be completely filled.
- *Timing instructions* restrict the execution window (*Market-on-close* orders are traded at closing prices; *Market-on-open* orders are traded on open prices)
- *Execution instructions* tell the broker how to arrange the trade (*Market-not-held* is a market order that the broker need not immediately execute or expose. The broker is expected to use discretion to find the best price).

[2] A *tick* is a change in price between trades. An *up-tick* is an increase in price; *down-tick* is a decrease in price.

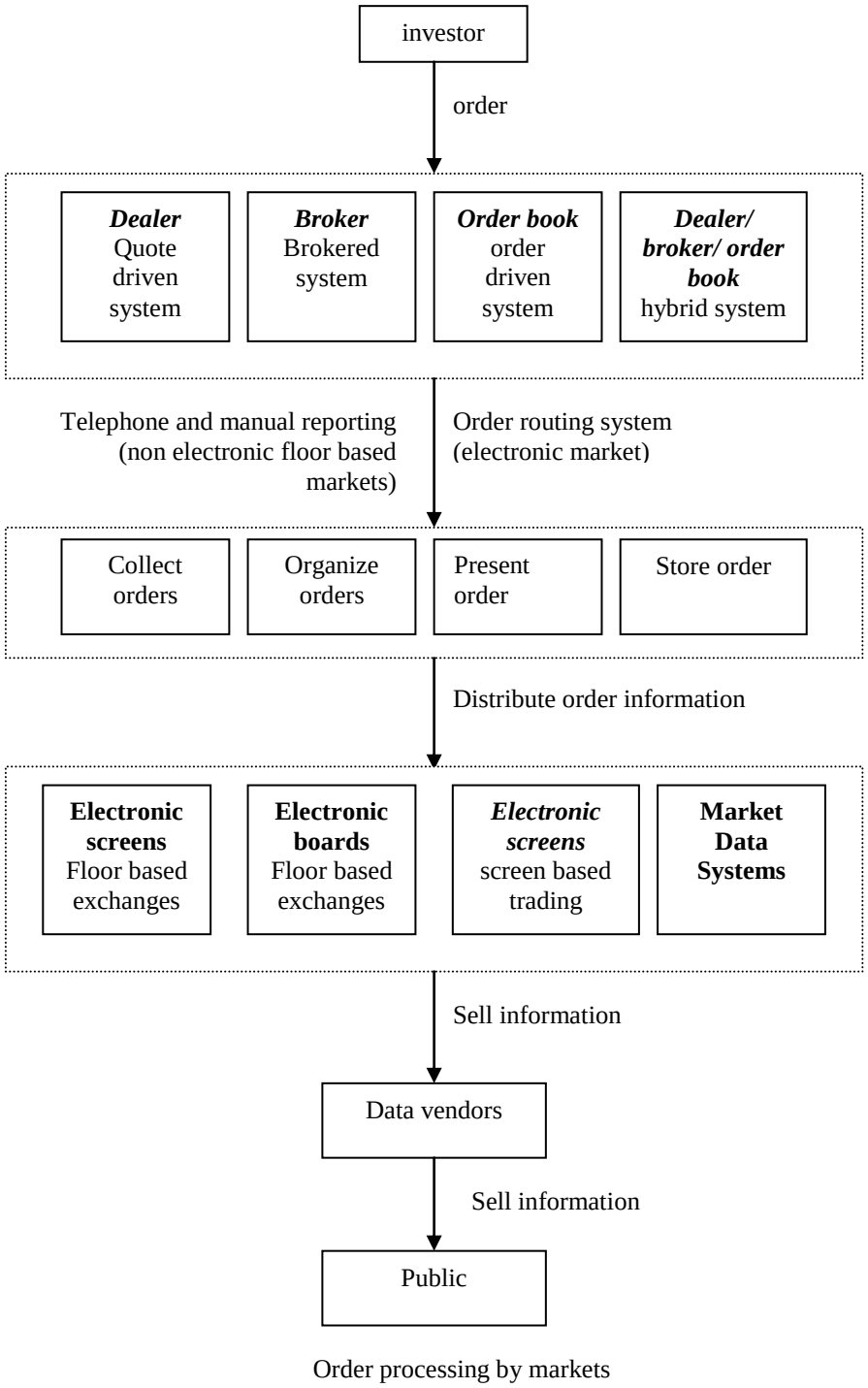


Figure A.3 Characteristics of the financial market structure

MARKET EFFICIENCY

Market efficiency is an important concern for both academics and practitioners. In academic research models efficient markets are prototyped, and tests for efficiency in different markets are undertaken. The efficient market hypothesis (EMH) introduced by Fama (1970 and 1991) is one of the central ideas in modern finance. There are different versions of the market efficiency hypothesis according to the information set that is assumed to be contained in market prices:

- weak form efficiency: current market prices reflect all information on past prices.
- semi-strong form efficiency: current market prices reflect all publicly available information.
- strong form efficiency: current market prices reflect both public and private information.

Malkiel (1996) stated that a capital market is efficient if it fully and correctly reflects all relevant information in determining security prices. This implies that it is impossible to make economic profits on the basis of that information set.

All the empirical research on the theory of efficient markets has been concerned with whether prices fully reflect particular subsets of available information. Weak form tests were conducted with the information subset of interest being past price histories [Fam70]. Weak form tests include serial correlation, runs, trading rules, and variance ratio tests [CLM97]. The semi-strong form efficiency test of the adjustment of prices to public announcements is conducted using an event study methodology. The test for private information (whether specific investors have information not in market prices) is used to test the strong form of efficiency [Fam91].

In practice, investors are highly concerned about getting fair prices and achieving high returns. Market efficiency implies also an optimal allocation of resources in the economy. Three types of market efficiency are identified [Arn98] and [Kol96]:

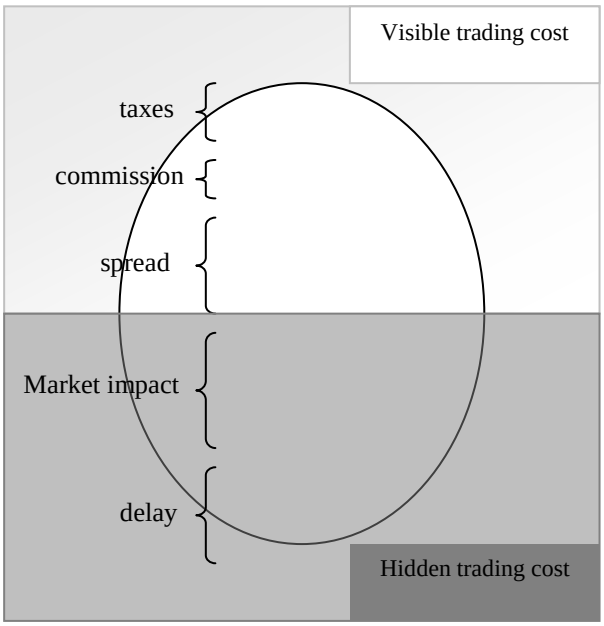
Operational efficiency: A financial market is operationally efficient if it works smoothly, with limited delays (orders can be transmitted from all parts of the world to a market very quickly, and are quickly executed and confirmed). Markets should carry out their operations at the lowest possible cost. Competition among markets is an ingredient in increasing operational efficiency. Technology is also an important factor in achieving operational efficiency. A market may be operationally efficient, however, without being informationally efficient.

Allocational efficiency: Resources in the economy are scarce, and it is important to allocate resources in a way to achieve optimum productivity. An efficient market should channel the fund to help in the growth of different industries.

Pricing efficiency / informational efficiency: In a pricing efficient market the investor can expect to earn a risk-adjusted return as prices move instantaneously and in an unbiased manner to any news. An informationally efficient market is one in which market prices adjust quickly in response to new information.

TRADING COST

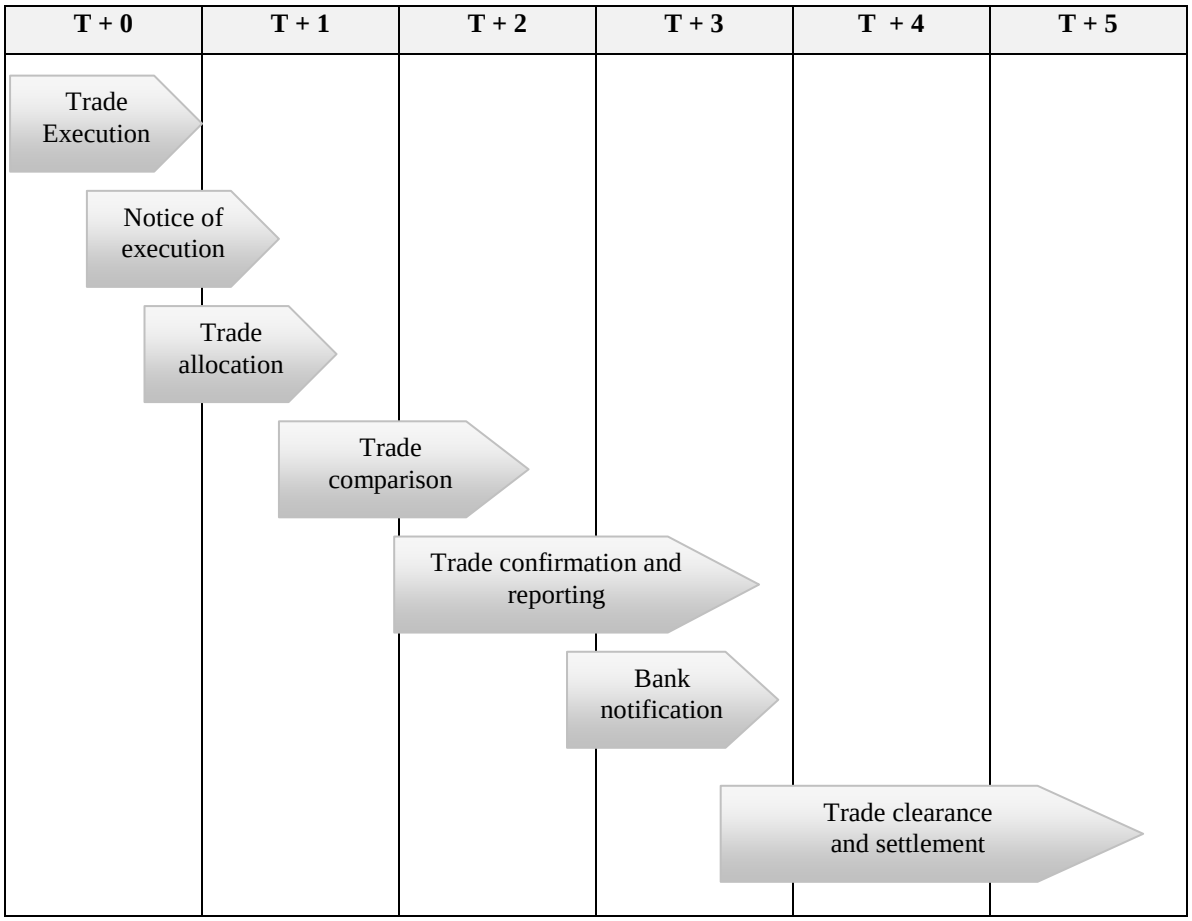
Haynes (2000) subdivided trading cost into a visible and hidden part. The visible part includes taxes, commission, and spread, while the hidden part of the trading cost includes market impact and delay.



Financial Trading Cost

STRAIGHT THROUGH PROCESSING

The five time steps to execute the three stages of the trading process in the old trading model [Mil99].



The old model for the trading process

EQUILIBRIUM PRICING RELATIONSHIPS

"To avoid having all your eggs in the wrong basket at the wrong time, every investor should diversify" Sir John Templeton

Modern Portfolio Theory (MPT) goes back to a revolutionary academic paper published by Harry Markowitz in 1952 [Mar59]. The concepts were developed by other American financial theorists such as William Sharpe in the 1950s and 1960s [RCO98]. The thinking behind this theory is that security diversification reduces risk. Spreading investment across a range of assets has a risk reducing effect. In a diversified portfolio unexpected bad news concerning one company will be compensated to a certain extent by unexpected good news about another. Markowitz argued that investors would optimally hold a mean-variance efficient portfolio, that is, a portfolio with the highest expected return for a given level of variance [CLM97]. Investors can select the optimum risk-return trade-off for themselves, depending on the extent of personal risk aversion.

Markowitz (1959) laid the groundwork for the CAPM. Sharpe (1964) and Lintner (1965) built on Markowitz's work to develop an equilibrium single period asset pricing model. They showed that if investors have homogeneous expectations and optimally hold mean-variance efficient portfolio then, in the absence of market frictions, the portfolio of all invested wealth, or the market portfolio, will itself be a mean-variance efficient portfolio. The usual CAPM equation is a direct implication of the mean-variance efficiency of the market portfolio [CLM97]. With the assumption of the existence of lending and borrowing at a risk free rate of interest the expected return of asset i is given by the equation:

$$E[R_i]=R_f + \beta_{im}(E[R_m]-R_f) \qquad \beta_{im} = \frac{Cov[R_i, R_m]}{Var[R_m]}$$

Where R_m is the return on the market portfolio, and R_f is the return on the riskfree asset.

The CAPM implies that the relationship between expected return and β_i is linear, only β_i is necessary to explain differences in returns among securities, the expected return of an asset with a β of zero is r_f, and the expected return of an asset with a β of one is the same as the expected return on the market [Abh2000-4].

$$E(R_i) = \alpha_0 + \alpha_1 \beta_i$$

Early tests of the CAPM are conducted by Lintner (1965), Black, Jensen and Scholes (1972). Roll's (1977) critique for the inconsistency of the empirical test of the CAPM, is that the market portfolio used in these tests is not the "true" market portfolio. Roll's main point was that the only potentially testable hypothesis associated with the CAPM is that the true market

portfolio is a mean variance efficient market portfolio. All other hypotheses (like expected return and beta being linearly related) are redundant given this main hypothesis. The Bad News was that this Main Hypothesis could not be tested since the Market Portfolio (the portfolio of all the assets in the economy) could not be observed [Abh00-4]. The CAPM is widely used project valuation, the evaluation of portfolio managers, and in determining the cost of capital [Abh00-5].

In the absence of a risk-free asset, Black (1972) derived a more general version of the CAPM, known as the Black version of CAPM [CLM97].

The Arbitrage Pricing Theory (APT) was introduced in Ross (1976) as an alternative to the CAPM [CLM97]. The APT allows for multiple risk factors and does not require identification of the market portfolio. This theory assumes competitive, frictionless markets, and can be stated as follows:

Suppose we have N assets, the return on these assets is an is an (N×1) vector with

$R = (R_1 R_2 \ldots R_N)^T$,

A is a (N×1) vector (the intercept of the factor model),

B is a (K×1) vector of factor sensitivities for asset i,

F is a (K×1) vector of common realizations,

and ε is an (N×1) disturbance term.

For the system of N asset the APT model is as follows:

$$\mathbf{R = a + Bf + \varepsilon}$$

Given this structure, Ross (1976) shows that the absence of arbitrage in large economies implies that

$$\mu \approx \iota\lambda_0 + B\lambda_k$$

where μ is the (N×1) expected return vector, λ_0 is the model zero- beta parameter and is equal to the riskfree return if such an asset exists, and $\boldsymbol{\lambda_K}$ is the (K×1) vector of risk premia.

In this model the market portfolio is one factor and additional factors are the state variables. The intuition for the additional factors is that these arise from investors demands to hedge uncertainty about future investment opportunities [Abh00-7].

FUND PERFORMANCE MANAGEMENT

A fund is a pool of money operated by a fund manager. There are many types of funds including: mutual funds, insurance funds, pension funds, or bank-pooled funds. The fund manager's job is to maximize the fund's returns at the least risk possible [NASD99]. The most common type of fund is the mutual fund which is an investment company that pools money from shareholders and invests in a diversified portfolio of securities [Inv98].

Trillions of dollars are invested in stocks world-wide by institutional portfolio managers. From a social perspective it is important to know whether these investors as a group add value to the portfolios they manage or whether they merely generate wasteful transaction costs through their active management. At the micro level it is important to know how to select a portfolio manager with the ability to add value to the portfolio he manages. Performance evaluation is a topic in financial economics that seeks to address both of these issues. In particular, it studies whether superior returns can be generated by active managers who are better able to collect and interpret information that help forecast securities returns. To evaluate whether a manager has generated superior returns we need to adjust his portfolio return for risk ([Gri95], [Abh00-6]). Two types of performance measurement can be conducted. The first involve the observation of the returns of the evaluated portfolio as well as the returns of a benchmark that consists of one or more portfolios along with a risk-free asset. The second type of performance measurement utilizes information about the composition of the evaluated portfolio but does not necessarily require a benchmark portfolio. Some performance measurement relies on the assumption of stationarity and normality of returns.

TRADING STRATEGIES

There are two basic orientations in trading strategies. A ***fundamental*** trader tries to predict what prices will do on the basis of factors that can be considered as affecting supply and demand. In the case of investment in commodities, the investor looks at future change in consumption patterns and the factors affecting the production of the commodity. In the case of currencies or financial futures, the investor monitors the change in political or economic policies. For stocks, the investor looks at the company's annual report, the quality of its management, and the factors affecting the changes in supply and demand for the services and goods required by and produced by the company. A ***technical*** trader looks for patterns in the trading data for the stock or commodity and tries to use them to predict the direction of future price changes. Typical criticism of fundamental trading would be: "if you can predict the changes in supply or demand that are going to occur, so can other traders. If they can predict

them, then the price probably already reflects these coming changes." Typical criticism of technical trading would be: "All your systems can do is tell you what happened in the past. They can't predict the future. History doesn't really repeat itself." The real difference between the two approaches is a philosophical one. The fundamental trader sees the behavior of the marketplace as the necessary result of supply and demand for the financial asset. Therefore prediction of price changes can best be done by correct analysis of factors influencing supply and demand. The technical trader sees his task as the prediction of trading behavior which is the cause of price movement. Accordingly, future behavior can be predicted by looking at past behavior and identifying patterns [Gol88]. Technical trading strategies [Gol88] include:

- *moving averages*: this is the simplest and most commonly used technical system. It is simply a price following system. It averages the prices of the most recent five days. This is compared to the average calculated for the previous day. If the average has gone up, the prices has gone up and you might want to be long, and if the prices has gone down, the prices are tending down and you might want to be short.
- *double moving averages*: this is similar to the moving averages except that two averages based on different numbers of days are used. When the average based on the smaller number of days rises or remains above the average based on larger number of days, it indicates that a price rise has begun or is accelerating. The reverse implication also holds true.
- *oscillator or momentum indicator*: Instead of comparing yesterday's average with today's average, this methods compares yesterday's change in moving average with today's change in moving average.
- *on-balance volume*: instead of relying only on price movement, this method considers also the volume of trading
- *stochastics*: the word stochastic is generally used in reference to random series. Stochastic technical trading tries to follow price movement by eliminating irrelevancies.

Computer programming for technical trading involves reading in the data, implementing the technical trading strategy, and devising two versions for going short or going long.

FINANCIAL INSTRUMENTS

The pace of financial innovation has accelerated sharply since the late 1970s, due to the deregulation of the financial industry and the increased competition and volatility in the financial markets. In the beginning, there were four instruments: a bank deposit, a bill of exchange (banker's acceptance), a bond, and equity. However, today a large number of new

financial products have been introduced with the aim of transferring risk, enhancing liquidity, generating credit, and generating equity. Walmsley (1988) classifies financial innovation as product/process and aggressive/defensive. Process innovation involves change in the process of financial markets such as the introduction of new trading technologies. Aggressive innovation is the introduction of a new product or process in response to perceived demand. A defensive innovation is introduced in response to changed environments or transaction costs. The effect and risks of recent financial innovation are ballooning trading volume, increasing exposure of financial institutions to each other, and a growing complexity in defining monetary aggregates and formulating monetary policy.

Broadly speaking, a financial security is a legal contract that confers the right to receive future benefits. A financial security is usually traded in organised markets. Financial securities are classified [Kol96, Elt95, CLM97, BKM96] as direct investment or indirect investments. Direct investments are classified as cash product vs. derivative security. Cash market instruments are classified as money or capital market instrument. Money market securities are short-term debt (maturity less than 1 year) issued by governments or companies. Examples include: Treasury Bills (TBs), Repurchase Agreements (REPOs), Certificates of Deposit (CDs), bankers acceptances, commercial paper, and eurodollars. Capital market instruments are long term securities with maturities greater than 1 year. They can be debt ("Fixed Income") or Equity.

Fixed income debts promise a stream of future cashflows as fixed interest payments ("coupons") paid at fixed dates, and repayment of the principal. Fixed income securities are issued by governments and companies.

The classical bond bears a fixed rate of interest, and matures on a date fixed at the time of issue. The price of a bond is the present value of its future cash flow. A bond is considered to be a fixed income security. One of the important characteristics of fixed income securities is the yield curve, which is a plot of fixed-income instruments against their maturities. A yield curve holds a lot of information. It displays the market's current yields for different maturities, it also shows the implied forecasts that the market is making about future rates.

An equity is a company share that gives its holder an ownership in the assets and earnings of a corporation. A share has no maturity, nor does it have any fixed claim on the assets or earnings. Shares give their holder rights to dividend if declared. The theoretical value of a stock is the present value or discounted value of the future stream of dividends. The stock today should be priced at the present value of dividend stream. Some ratios are also important in the valuation of equities. These are:

Price-earnings ratio = market price of share / net earning per share

Dividend yield = dividend per share / market price of share

Payout ratio = dividend per share/ earnings per share

A derivative instrument has a value *derived* from the value of some underlying asset (equity, bonds, currencies, ...), such as futures, options, swaps, etc...

Options on financial instruments, notably on stocks, have been around for years, but until 1973 they were tailor made. It was possible to buy an option but not to trade it. Later on, exchange-traded options were introduced: the first was the Chicago Board Options Exchange (CBOE). Basically, an option is an agreement between two parties, giving the right to buy/sell an instrument (stock, bond, future contract, interest rate, or foreign currency) under certain conditions. There are two basic types of options: *puts* and *calls*. A call option gives the buyer the right to buy or "call away" a specified amount of the underlying security at a specified price, during a specified period. The price at which the instrument may be bought is the *exercise price* or the strike price. The last date on which the option may be exercised is the *expiration date*. The price paid for the option is called the *premium*. Depending on the underlying instrument, the mechanics of options trading vary from contract to contract. Option valuation is quite complex. The price of an option is determined by six factors: (1) the price of the underlying instrument; (2) the striking price of the option; (3) the time remaining until expiration of the option; (4) the volatility of the underlying instrument; (5) the current, risk free interest rate; (6) the dividend rate of the underlying stock or the interest rate of the underlying security. The most widely used way for valuing options works on a principle known as the riskless hedge. The Black-Scholes model, used for equity option valuation, uses riskless hedge. The assumption of riskless hedge valuation models are:

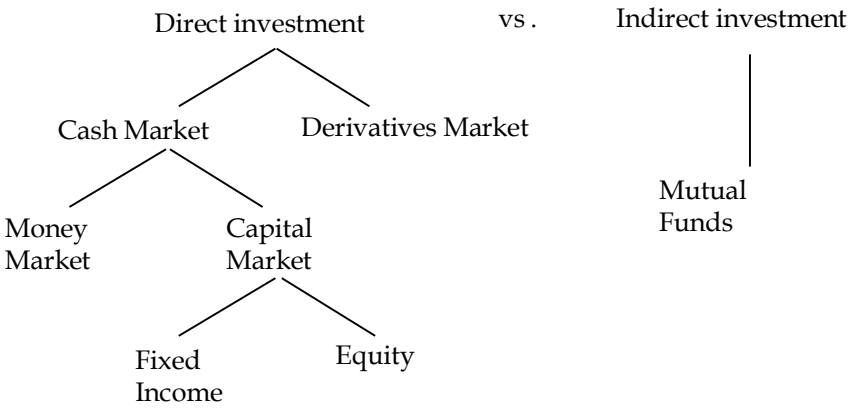
1) prices may change rapidly but cannot jump, and one can trade continuously in the market;
2) there is a risk free rate of interest for borrowing and lending from the current period until the expiration of the option;
3) transaction cost and taxes are ignored.

In the Black-Scholes formula the key assumption is that the price of the stock is a random variable in continuous time, and that the percentage change in its price has a normal distribution.

The above description of instruments covers direct investment. Indirect investment includes investment in Mutual Funds. These can be "open-end" funds (Unit Trusts), or "closed-end" funds (Investment Trusts). Mutual funds are funds operated by an investment company that

raise money from shareholders and invests it in stocks, bonds, options, commodities or money market securities. A closed-end fund has a limited numbers of shares outstanding. A closed end fund starts with a set number of shares that are traded on a stock exchange. Open end funds continually creates new shares on demand. Shareholders can redeem their shares at any time at the prevailing market price.

The following figure – based on Gruber (et al, 1995) - depicts the classification of financial instruments.



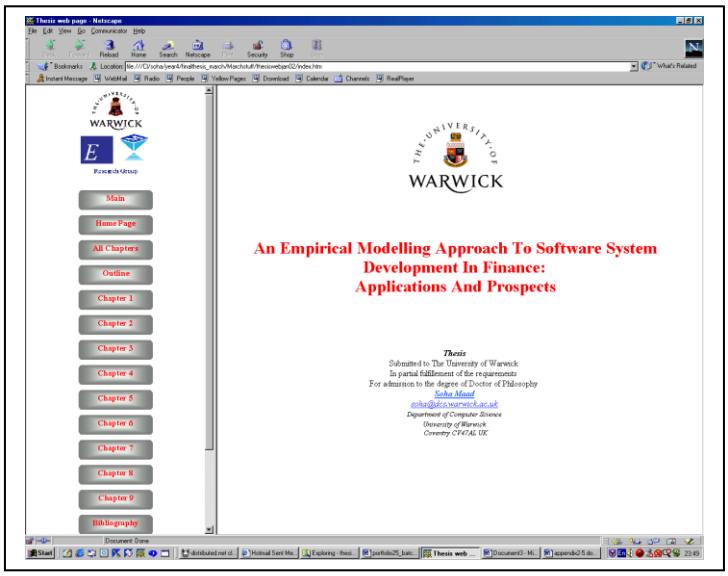
Classification of financial instruments

The thesis home page

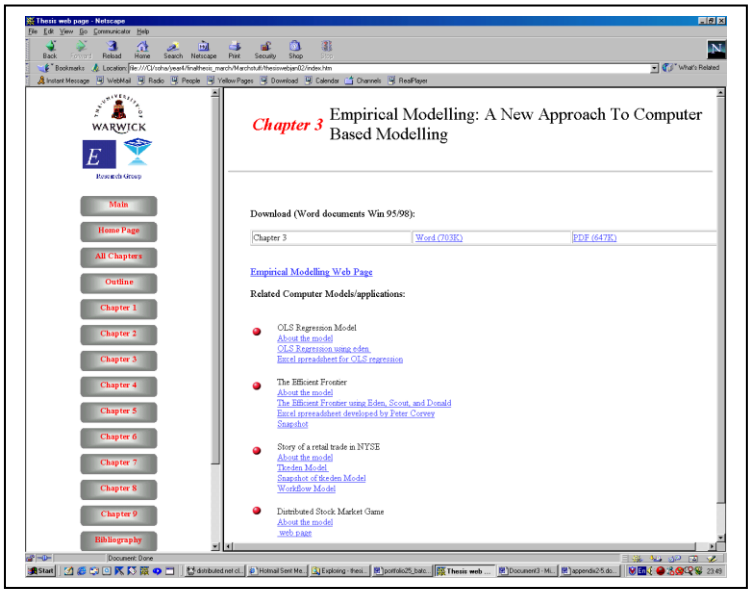
Home page of every chapter

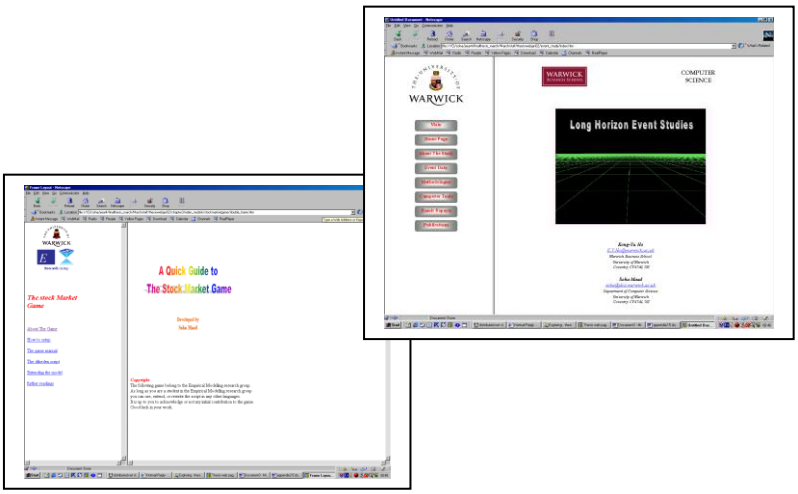
Thesis downloads

Personal information

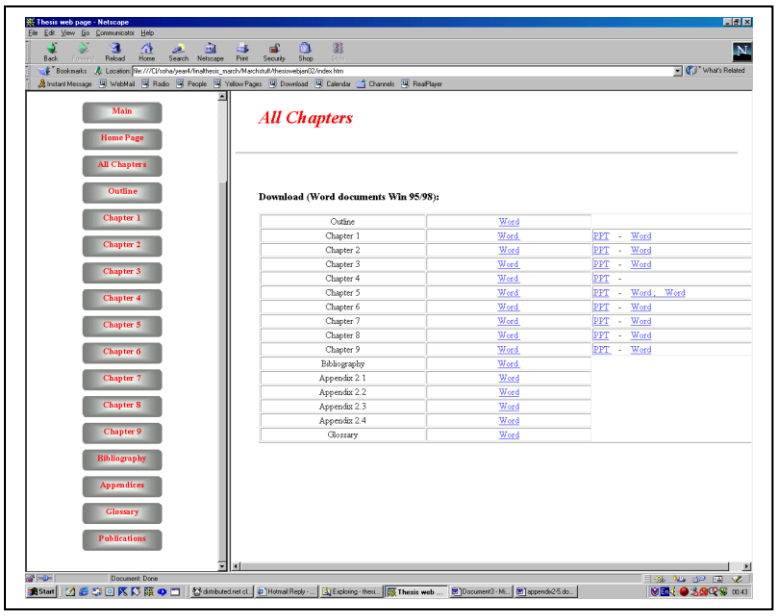


Home page of every chapter

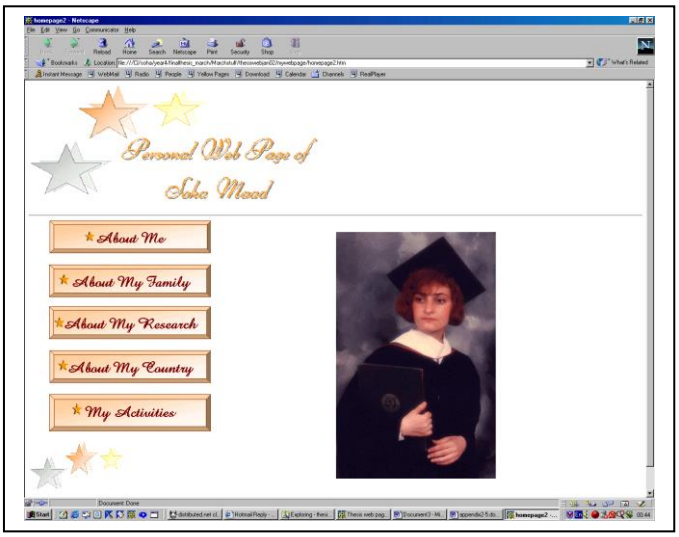


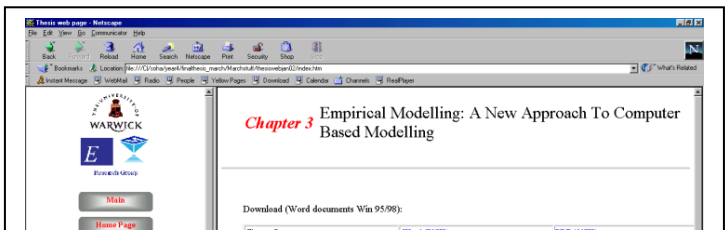


Thesis downloads



Personal information




Thesis web page - Netscape
WARWICK
Main
Home Page
Chapter 3
Empirical Modelling: A New Approach To Computer Based Modelling
Download (Word documents Win 95/98):

Glossary

Adverse selection cost: The cost incurred by a market maker (dealer) for taking part in trades with investors who may have superior information.

Artificial Intelligence: Artificial Intelligence (AI) is a field of study that investigates new uses of computers (hardware and software) that goes beyond routine calculation and operations to mimic human thinking. In defining the term Artificial Intelligence we can define the term 'artificial' and the term 'intelligence' separately or we can infer a meaning for the expression 'artificial intelligence'. A universally accepted definition [RK91] is "Artificial Intelligence is the study of how to make computers do things which at the moment, people do better". Artificial Intelligence does not address problems that cannot be solved by either the human being or computers.

Source: [RK91]

Bid-ask spread: The bid-ask spread is part of the transaction costs that investors will have to pay in order to take a position in securities markets [Chi94]. Bid-ask spreads are computed as the difference between the price at which market dealers are willing to sell (ask) and the lower price at which they are willing to buy (bid). From the point of view of market makers, bid-ask spreads are a source of compensation for maintaining high liquidity in the markets and, more specifically, to account for order processing costs, inventory costs, and adverse selection costs.

Source: [Chi94]

Business Process Modelling / Re-engineering: BPM / BPR seeks to devise new ways of organizing tasks, organising people, and making use of IT systems so that the resulting processes will better support the goals of the organization. The central tenets of BPR are: 1) radical change and assumption challenge; 2) process and goal orientation; 3) organisational restructuring; 4) the exploitation of enabling technologies, particularly information technology.

Source: [VRWH94]

Cognitive Technology: Cognitive Technology is concerned with the interaction between two worlds: that of the mind and that of the machine. The technologies of the third millennium … force us to re-examine fundamental concepts of embodiment and consciousness which frame our understanding of the relationship between minds and machines. Cognitive Technology addresses this issue using the diverse perspectives afforded by a wide range of disciplines, and evidence drawn from both contemporary developments and the history of technology.

Source: [CT01]

Data mining: It is the analysis of a large population of unknown data to identify hidden patterns or characteristics that might be useful in delivering business value.

Source: [Mat99].

Data Warehouse: (1) A subject oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process. A repository of consistent historical data that can be easily accessed and manipulated for decision support. (2) An implementation of an informational database used to store sharable data sourced from an operational database of record. It is typically a subject database that allows users to tap into a company's vast store of operational data to track and respond to business trends and facilitate forecasting and planning efforts.

Source: [FS99]

Database Technology: A database is a permanent self descriptive store of data. It contains the data structure (schema) and the data. A database manager is the software managing access to the database. Different database paradigms exist: relational (where data is perceived as tables, and data is accessed by a relational database management system), multidimensional (tailored for analytical applications with complex relationships between data), object oriented (based on the concept of embodied objects, and relationships between objects established through pointers).

Source: [Bla01]

Efficient Market Hypothesis: The efficient market hypothesis (EMH) introduced by Fama (1970 and 1991) is one of the central ideas in modern finance. There are different versions of the market efficiency hypothesis according to the information set that is assumed to be

contained in market prices: in weak form efficiency current market prices reflect all information on past prices; in semi-strong form efficiency current market prices reflect all publicly available information; in strong form efficiency current market prices reflect both public and private information.Malkiel (1996) stated that a capital market is efficient if it fully and correctly reflects all relevant information in determining security prices. This implies that it is impossible to make economic profits on the basis of that information set.

Source: [Fam70, Fam91, Mal96]

Ethnomethodology: the empirical investigation ('-ology') of the methods ('method') people ('ethno') use to make sense of and at the same time accomplish communication, decision making, reasonableness, and action in every day life.

Source: [Rog83]

Financial Security: A financial security is any contract that derives its value from uncertain future events. This economist's definition encompasses common stocks, preferred stocks, bonds, convertible bonds, warrants, options, future contracts, forward contracts, foreiggn exchange contracts, swaps, re-insurance contracts, and even many betting contracts.

Source: [Har98]

Human Computer Interaction: Human computer interaction (HCI) is inherently a multidisciplinary subject. It involves theories of human behaviour as well as the principles of computer systems design [PRSBHC94]. Exploration of the social and organizational context of the user is also important. Research in the area of HCI aims at developing the right system to fit a specific purpose, it involves studying users and their tasks and relating this information to design styles, human factors theories, guidelines and standards in order to select an appropriate form of interaction. Computer supported cooperative working (CSCW), hyper and multimedia and virtual reality are state of the art in applications in the area of HCI. It is commonly argued that future human-computer interaction will come to resemble our everyday interaction with the world [BRSW98].

Source: [PRSBHC94, BRSW98]

Investment: The term investment can take two meanings depending on who is initiating it. An investor may purchase a financial asset with a view to make long term return from

appreciation in the price of the asset. This is equivalent to saving money in a form of asset, associated with it is a transfer of ownership of this asset. On the other hand companies or governments can raise funds in the financial market (borrow from savers). This is known as capital investment by companies or governments.

Market capitalization: It is the number of traded shares times the trading volume. A large market capitalization is usually a good sign of an active financial market.

Market Microstructure: Market microstructure is the academic name for the branch of financial economics that investigates trading and the organization of securities markets. This important field of study has grown substantially since the Stock Market Crash of 1987.

Source: [Har98]

Number of companies listed : This refers to the number of companies registered and listed in the financial market. A large number of listed companies might appear to be a good sign of an active financial market. However, this indicator alone cannot tell much about the characteristics of the financial market if not accompanied by the traded volume per share of a listed company. Some firms listed in emerging financial markets have a low trading volume and even no trading volume.

Object Oriented Technology: Object Oriented technology uses the concept of objects, the theory of abstract data types, and the concept of message passing and inheritance to represent real world entities and model complex systems.. The concept of object was introduced in [DN66]. The theory of abstract data types was formalized in [LZ75]. The concept of message passing and inheritance were introduced in [HBS73]. The linking of object by the “is-a” relation was introduced in [Win84].

Source: [DN66, LZ75, HBS73, Win84]

Portfolio: a combination of assets

Regression analysis: It is used to analyse a population and identify the nature of the relationship between two or more variables.

Source: [Mat99]

Software Engineering: In the belief that software design, implementation, and maintenance could be put on the same footing as traditional engineering disciplines, a NATO study group in 1967 coined the term software engineering. This term was endorsed by the

NATO software engineering conference held in Germany. The software crisis, manifested by low quality, excessive cost, and bypassed delivery time deadlines, motivated the need of the adoption of philosophies and paradigms of established engineering discipline. Software engineering involves the application of engineering principles to software system development to deliver a high-quality software product. This involves the application of scientific principles to transform a problem into a working software solution, and the subsequent maintenance of the software until the end of its useful life. Software engineering is more than just programming; the software engineering process starts long before programming and continues long after the code is written.

Source: [Sch93, Dav93-A]

Software integration: the broad definition of integration "the coherent merging of entities, having different behaviours and attributes, to obtain a unified entity that can realise the behaviours of its components and whose attributes are derived from but are not necessarily the same as those of its components. Integration might be necessary to accommodate a new style of relationship between different entities, or it might be optional with the aim of enhancing performance and gaining value-added advantages." applies to software integration.

Source: [BM99]

Stock index: A stock index is a measure of the performance of a hypothetical basket of stocks. Only the basket is hypothetical: the stocks are real and their performance is real. The range of indices continues to grow. Some are based on all the stocks on a particular exchange, others on a sampling of them, still others on all or a sampling of those in a particular industry. A stock index can serve a variety of purposes. The most general purpose is probably as a measure of changes in the economy. As a corollary of this, it can be used to compare the relative performances of the economy as a whole and a particular industry. It can be used also to compare the performance of one country relative to another.

Source: [Gol88]

Virtual Reality: The term "virtual reality" is credited to Jaron Lanier, founder of VPL Research. Earlier experimenters, like Myron Krueger in the mid-1970s, used the term artificial

reality [Kru83, MS94]. Krueger had an ambitious use of the term artificial reality: “a world created by human ingenuity and embracing mysterious cosmic forces and nature’s power”. Machover et al (1994) definition of Virtual Reality is : *"Virtual Reality is the technology that facilitates the operation of complex systems, consuming information and turning it into knowledge (that most valuable of human resources) "*. Landauer et al (1997) defines Virtual Reality as " *the presentation of computer-simulated worlds to the human senses*".

Source: [MS94, MT94, LBBRS97, Kru83]